НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Інститут телекомунікаційних систем

Кафедра Телекомунікаційних систем

«На правах рукопису»
УДК _____

«До захисту допущено»
Завідувач кафедри
_____ Л.О. Уривський
«___»_____20__ р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 172 Телекомунікації та радіотехніка

на тему: «Дослідження сучасних алгоритмів побудови цифрових водяних знаків для відео-контенту»

Виконала:
студентка ІІ курсу, групи ТС-61м
Кулик Марина Вікторівна _____

Керівник:
к.т.н., старший викладач
Лісковський Ігор Олегович _____

Рецензент:
Посада, науковий ступінь, вчене звання,
Прізвище, ініціали _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2018 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут телекомунікаційних систем
Кафедра Телекомунікаційних систем

Рівень вищої освіти – другий (магістерський) за освітньо-науковою програмою

Спеціальність (спеціалізація) – 172 «Телекомунікації та радіотехніка» (172.3620.1 «Телекомунікаційні системи та мережі»)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Л.О. Уривський
«___»_____20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Кулик Марині Вікторівні

1. Тема дисертації «Дослідження сучасних алгоритмів побудови цифрових водяних знаків для відео-контенту», науковий керівник дисертації Лісковський Ігор Олегович, к.т.н., старший викладач, затверджені наказом по університету від «___»_____ 20__ р. №____

2. Термін подання студентом дисертації _____

3. Об’єкт дослідження цифровий водяний знак (digital water mark).

4. Предмет дослідження алгоритми внесення цифрових водяних знаків у контент та методи робастного вбудовування ЦВЗ.

5. Перелік завдань, які потрібно розробити:

- розглянути основні механізми внесення цифрових водяних знаків у контент: внесення цифрових водяних знаків у зображення, звукові файли, відео;
- виробити критерії для алгоритму вбудовування ЦВЗ;
- порівняти основні методи вбудовування ЦВЗ і вибрати відповідний;

- розробити алгоритм захисту мультимедійних файлів на прикладі захисту зображень з подальшою можливістю розширення алгоритму для відеоконтенту;

- дослідити інструментарій для реалізації алгоритму та представити прототип програмної реалізації алгоритму;

- протестувати роботу алгоритму в умовах, наближених до реальних.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу

Плакат №1 «Тема, мета та завдання магістерської дисертації»

Плакат №2 «Актуальність та постановка задачі»

Плакат №3 «Області застосування стеганографії»

Плакат №4 «Компроміс при виборі характеристики стеганосистеми»

Плакат №5. «Узагальнена класифікація стеганографічних методів»

Плакат №6. «Алгоритм вбудовування цифрового водяного знаку»

Плакат №7. «Результати тестування розробленого алгоритму»

Плакат №8. «Висновки»

7. Орієнтовний перелік публікацій

1. Лісковський І.О., Кулик М.В. Реалізація стеганографічної системи для відео з використанням сингулярного розкладання. // Дванадцята міжнародна науково-технічна конференція «Проблеми телекомунікацій»; Десята міжнародна науково-технічна конференція студентів та аспірантів «Перспективи розвитку інформаційно-телекомунікаційних технологій та систем»: Матеріали конференції. К.: НТУУ "КПІ", 2018. – с. 288-290.

2. Кулик М.В. Реалізація алгоритму робастного внесення цифрових водяних знаків мовою Python / Кулик М.В. // Міжнародна науково-практична конференція «Наука та освіта: ключові питання сучасності»: зб. наук. праць «λόгοσ». – 2018.

8. Дата видачі завдання 10 вересня 2016 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Розробка, оформлення, узгодження та затвердження технічного завдання на роботу. Аналітичний огляд інформаційних матеріалів. Підбір та опрацювання необхідної науково-технічної літератури.	01.09.2016- 31.12.2016	
2	Огляд областей застосування стеганографії та вивчення основної термінології стеганографії.	10.01.2017 - 29.02.2017	
3	Розгляд механізмів внесення цифрових водяних знаків в контент. Внесення цифрових водяних знаків в зображення, відео, звукові файли.	01.03.2017 – 30.07.2017	
4	Пропозиції щодо механізму внесення цифрових водяних знаків в відео-контент.	01.08.2017 – 31.10.2017	
5	Розробка алгоритму внесення ЦВЗ та алгоритму формування мітки.	01.11.2017 – 30.01.2018	
6	Практична реалізація алгоритму вбудовування та вилучення цифрового водяного знаку з відеоконтенту.	01.02.2018 – 31.03.2018	
7	Тестування розробленого алгоритму та аналіз отриманих результатів.	01.04.2018 – 30.04.2018	
8	Узагальнення і оцінювання результатів досліджень, підготовка підсумкового звіту. Подання роботи до приймання, та її захист.	01.05.2018 - 20.05.2018	

Студент Кулик М.В.

Науковий керівник дисертації Лісковський І.О.

РЕФЕРАТ

Обсяг магістерської дисертації складає 95 сторінок, зокрема 24 ілюстрації, 1 таблицю, 13 формул та 41 джерело інформації.

Актуальність теми. Зростання кількості контенту у вигляді мультимедійних файлів і, як наслідок, зростання потреби в забезпеченні захисту авторського матеріалу, що охороняється законами про захист авторського права, пред'являє нові вимоги до технічних засобів, які такий захист можуть забезпечити. У даній роботі пропонується алгоритм стеганографічного перетворення, ключовими характеристиками якого є швидкість роботи, висока стійкість стего-контейнера до атак і можливість вилучення внесеної мітки без наявності контейнера-оригіналу.

Тема магістерської дисертації є актуальною, тому що більшість авторських творів, які можна вільно знайти в мережі Інтернет, знаходяться саме з порушенням прав інтелектуальної власності, проблема несанкціонованого тиражування контенту нікуди не зникла, а просто видозмінилася - тепер замість копіювання купленого компакт-диска люди копіюють сам контент, доступ до якого продає компанія. В інтересах компаній перешкоджати такому незаконному розповсюдженню або, принаймні, вчасно виявляти джерело «витоку» контенту й обмежувати доступ клієнта до ресурсу. Основною ідеєю цієї роботи і є захист цифрового контенту.

Метою випускної кваліфікаційної роботи є розробка методу робастного захисту авторських мультимедійних файлів алгоритмами цифрової стеганографії на прикладі вбудовування ЦВЗ в зображення з можливістю подальшого розширення для відеоконтенту.

Відповідно до поставленої мети були сформульовані такі завдання:

- розглянути основні механізми внесення цифрових водяних знаків у контент: внесення цифрових водяних знаків у зображення, звукові файли, відео;

- виробити критерії для алгоритму вбудовування ЦВЗ;

- порівняти основні методи вбудовування ЦВЗ і вибрати відповідний;

- розробити алгоритм захисту мультимедійних файлів на прикладі захисту зображень з подальшою можливістю розширення алгоритму для відеоконтенту;

- дослідити інструментарій для реалізації алгоритму та представити прототип програмної реалізації алгоритму;

- протестувати роботу алгоритму в умовах, наближених до реальних.

Об’єктом дослідження є цифровий водяний знак (digital watermark).

Предметом дослідження є алгоритми внесення цифрових водяних знаків у контент та методи робастного вбудовування ЦВЗ.

Методи дослідження. В ході роботи були використані: методи теоретичного дослідження, емпіричний підхід, методи логічного проектування та процедурної алгоритмізації, прийоми об'єктно-орієнтованого та логічного програмування.

Наукова новизна отриманих результатів. В результаті виконання дисертаційного дослідження був розроблений алгоритм швидкого робастного вбудовування ЦВЗ для зображень, що дозволяє вилучати приховане повідомлення без наявності контейнера-оригіналу.

Практичне значення отриманих результатів. При виконанні роботи був реалізований прототип у вигляді скриптового Python-додатку, доступного для запуску з консольного рядка за допомогою інтерпретатора Python версії 2.7 і вище.

Апробація результатів дисертації. Основні результати дисертаційного дослідження оприлюднено в ході 2 наукових конференцій, серед яких:

- Дванадцята міжнародна науково-технічна конференція "Проблеми телекомунікацій", 2018 р. (ІТС, НТУУ “КПІ ім. Ігоря Сікорського”);
- Міжнародна науково-практична конференція «Наука та освіта: ключові питання сучасності», 2018. (м. Чернігів)

Публікації. Основні положення і результати дисертаційної роботи знайшли своє відображення у 2 публікаціях: публікацій матеріалів і тез доповідей на Дванадцятій міжнародній науково-технічній конференції "Проблеми телекомунікацій", 2018 р. (ІТС, НТУУ “КПІ ім. Ігоря Сікорського”) та на Міжнародній науково-практичній конференції «Наука та освіта: ключові питання сучасності», 2018. (м. Чернігів).

Ключові слова: *цифровий водяний знак, стеганографія, прихована передача, інформаційна безпека, захист даних.*

ABSTRACT

The work contains 95 pages, 24 illustrations, 1 table, 13 formulas and 41 sources.

Relevance of the topic. The growth in the number of produced content in the form of media data and, as a result, the growing need to protect copyrighted material protected by copyright laws poses ever new requirements for technical means that such protection can provide. In this paper, we propose a method and algorithm for steganographic transformation, the key characteristics of which are the speed of work, high stability of stego for attacks (robustness), and the ability to extract a recorded mark without the presence of an original container.

The topic of the master's thesis is relevant, because the majority of all copyrights that we can freely find on the Internet are in violation of intellectual property rights., so the problem of unauthorized copying has not disappeared, but simply changed - now instead of copying a purchased CD people copy the content itself, the access to which the company sells. It is in the interest of companies to prevent such unlawful proliferation, or at least, in a timely manner, to identify the source of the "leakage" of content and to limit access of the client to the resource. The main idea behind this work is the protection of digital content.

The purpose of the thesis is to develop a method of robust protection of author's multimedia files by algorithms of digital steganography, for example, embedding the digital watermark in an image with the possibility of further expansion for video content.

In accordance with the stated goal, the following objectives were formulated:

- to consider the basic solutions in the field of digital and computer steganography;

- to consider the main mechanisms for the embedding of digital watermarks in the content: embedding of digital watermarks in the image, audio files and video;

- to develop criteria for the embedding algorithm;

- to compare the basic methods of embedding the digital watermarks and choose the appropriate one;

- develop an algorithm for protecting multimedia files on an example of image protection with the further ability to expand the algorithm for video content;

- to do research of a toolkit for implementation of the algorithm and to present the prototype of the program realization of the algorithm;

- test the work of the algorithm in conditions close to the real.

The object of research is a digital watermark.

The subject of research are algorithms for embedding digital watermarks in content and methods of robust digital watermark embedding.

Research methods. In the course of the work were used: methods of theoretical research, empirical approach, methods of logical design and procedural algorithmization, techniques of object-oriented and logical programming.

Scientific novelty of the obtained results. As a result of the dissertation research, an algorithm for rapid robust embedding of the digital watermark for images was developed, which allows you to remove a hidden message without the presence of the original container.

The practical significance of the results. In the course of the work, a prototype in the form of a script Python application that was available to run from the console line using the Python interpreter version 2.7 and above was implemented.

Approbation of the results of the dissertation. The main results of the dissertation research were published during 2 scientific conferences, among them:

- Twelfth International Scientific and Technical Conference "Problems of Telecommunications", 2018 (ITS, NTUU "KPI named after Igor Sikorsky");

- International scientific and practical conference "Science and education: key issues of our time", 2018. (Chernihiv)

Publications. The main provisions and results of the dissertation were reflected in 2 publications: publications of materials and abstracts at the Twelfth International Scientific and Technical Conference "Problems of Telecommunications", 2018 (ITS, NTUU "KPI named after Igor Sikorsky"); International scientific and practical conference "Science and education: key issues of our time", 2018. (Chernihiv).

Keywords: *digital watermark, steganography, hidden transmission, information security, data protection.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

DVD	(Digital Video Disc) — носій інформації у вигляді диска, який має можливість зберігати інформацію за рахунок використання лазера.
GUI	(Graphical User Interface) — тип інтерфейсу, який дозволяє користувачам взаємодіяти з електронними пристроями через графічні зображення та візуальні вказівки
JPEG	(Joint Photographic Expert Group) – растровий формат збереження графічної інформації, що використовує стиснення з втратами
MPEG	(Moving Picture Experts Group) - стандарт, який переважно використовується для кодування відео й аудіо при мовленні, включаючи супутникове мовлення і кабельне телебачення
MSE	(Mean Square Error) – середньоквадратична похибка
PNG	(Portable Network Graphics) - растровий формат збереження графічної інформації, що використовує стиснення без втрат.
RGB	(Red, Green, Blue - червоний, зелений, синій) - адитивна колірна модель, що описує спосіб синтезу кольору, за якою червоне, зелене та синє світло накладаються разом, змішуючись у різноманітні кольори
SNR	(Signal-to-noise ratio) - відношення сигнал/шум
ДКП	дискретне косинусне перетворення
ДПФ	дискретне перетворення Фур'є
ЗСЛ	зорова система людини
НЗБ	найменш значущий біт
ПВП	псевдовипадкова послідовність

ПЗ	програмне забезпечення
ЦВЗ	цифровий водяний знак
ЦОС	цифрова обробка сигналів
ШДПФ	швидке дискретне перетворення Фур'є

ВСТУП

З кожним роком кількість наукових публікацій, присвячених проблемам стеганографії та стеганоаналізу невпинно зростає. Основними джерелами для вітчизняних дослідників є праці В.Г. Грибуніна, А.В. Аграновського, Г.Ф. Конаховича, В.К. Задираки, М.Є. Шелеста, В.О. Хорошко, О.Д. Азарова, М.Є. Шелеста, Ю.Є Яремчука, О.В. Генне, В.С. Барсукова, А.П. Романцова, Е. Ю. Мерзлякової, А.Т. Алиева та ін. Серед зарубіжних вчених, що зробили значний внесок у формування та подальший розвиток стеганографії та стеганоаналізу можна виділити: Д. Фрідріч (J. Fridrich), М. Кун (M.G. Kuhn), С. Кравер (S.Craver), Ш. Каценбейзер (S. Katzenbeisser), А. Фестфельд (A. Westfeld), Е. Кох (E. Koch), Ц. Жао (J. Zhao), К. Лю (Q. Liu), Ф. Петикола (F. Petitcolas) та ін.

Останнім часом, буквально останнє десятиліття, спостерігається стійка тенденція до перенесення різних інформаційних сервісів з персональних комп'ютерів, робочих станцій і серверів в так звану «хмарну» («cloud») - інфраструктуру, що підтримується сторонніми компаніями й доступна глобально. Якщо раніше люди для домашньої самоосвіти купували книги й компакт-диски, то в теперішній час, коли високошвидкісний доступ в інтернет є у багатьох, необхідність в цьому відпала, тому що вся інформація доступна з мережі, в тому числі програми для навчання, текст, відеоуроки, презентації та інше.

Розвиток мережевої інфраструктури призвів до того, що до персонального комп'ютера тепер все рідше пред'являються вимоги великої обчислювальної потужності або великого об'єму жорсткого диска, адже всі дані зберігаються віддалено і багато «складних» операцій так само виконуються на віддалених серверах. Прикладів, коли компанії відмовляються від підтримки вбудованих «standalone» рішень на користь хмарних аналогів - маса. Такі рішення та сервіси, що доступні глобально з будь-якої точки світу в будь-який момент часу і підтримують синхронізацію даних між різними клієнтами, називаються SaaS-платформами, що розшифровується як Software-as-a-Service [1]. Таким чином, компанії тепер найчастіше заробляють не на продажу копій свого програмного забезпечення, а на

наданні користувачам і компаніям віддаленого доступу до цих програм. У цього підходу є свої плюси й мінуси, але сучасне суспільство оцінило переваги цієї моделі й такі сервіси стали з'являтися все частіше.

Однак, проблема несанкціонованого тиражування контенту нікуди не зникла, а просто видозмінилася - тепер замість копіювання купленого компакт-диска люди копіюють сам контент, доступ до якого продає компанія. В інтересах компаній перешкоджати такому незаконному розповсюдженню або, принаймні, вчасно виявляти джерело «витоку» контенту й обмежувати доступ клієнта до ресурсу. Основною ідеєю цієї роботи є захист цифрового контенту, була поставлена мета розробити алгоритм, швидкість роботи якого дозволить використовувати його в веб-проектах, тобто ставиться вимога роботи алгоритму практично в режимі реального часу. При цьому характеристики робастностного запису ЦВЗ повинні дозволяти читати водяний знак навіть після стиснення оригіналу в форматі PNG в формат JPEG.

Наведемо сценарій використання розроблюваного алгоритму для обґрунтування практичної значущості роботи.

Існують компанії, бізнес яких зосереджений виключно на наданні інформаційних послуг. Інтернет для таких компаній є основним каналом зв'язку з клієнтом і основним каналом поширення своєї продукції. Наприклад, серед таких можна відзначити електронні ЗМІ, форуми, блоги, освітні ресурси, новинні підписки та інші.

Для зручності, зупинимося на прикладі освітніх ресурсів.

Серед користувачів глобальної мережі Інтернет все більш популярним стає явище онлайн-навчання - спосіб, при якому ви зможете отримати корисні прикладні вміння або теоретичні знання, не виходячи з дому. Для цього потрібно бути учасником будь-якого освітнього ресурсу, який публікує курси з цікавими для вас тем. Такі курси створюються силами приватних компаній і поширюються по моделі платних підписок або одноразової оплати доступу до курсу (наприклад, Prometheus, Udemy, Coursera, тощо).

Таким чином, автори передають свої знання і досвід студентам в обмін на винагороду. Це дуже хороший спосіб навчитися чомусь новому, що не входить у стандартні академічні програми або з'явилося не так давно, а ринок уже вимагає фахівців відповідного профілю. Наприклад, це дуже актуально в сфері IT, зокрема веб-розробки - back-end і front-end технології розвиваються зараз дуже стрімко, не без причини широкого поширення хмарних технологій, і з'являються нові фреймворки, бібліотеки, програмне забезпечення, роботі з якими не навчить жоден університет, а компаніям вже зараз потрібні фахівці, здатні впроваджувати найновіші розробки.

Ще як приклад можна навести те, що існують компанії, що пропонують свої унікальні методики самостійного вивчення англійської мови за допомогою проходження авторських онлайн-курсів. Це може бути корисно людям у віддалених куточках країни; людям, у яких немає достатньої кількості часу для очного навчання в групах, і просто всім тим, кому такий спосіб отримання інформації здається більш комфортним і привабливим.

На цьому грунті і виникають різні компанії-постачальники освітнього контенту, бізнес-модель яких будується на продажу доступу до курсів великій кількості студентів з різних куточків планети.

Тут і проявляється проблематика - захист авторського контенту. В інтересах компаній зробити так, щоб курс, проданий одному студенту, був доступний тільки цьому студенту. Якщо ж він поширює матеріали курсу (наприклад, відеоуроки) серед своїх знайомих або просто серед всіх бажаючих, то компанія зазнає збитків у вигляді втраченої вигоди, адже всі ті, хто отримав матеріали курсу від одного студента могли стати клієнтами компанії.

Явище несанкціонованого розповсюдження контенту, захищеного авторськими правами, називається комп'ютерним піратством. Комп'ютерне піратство - це незаконне розповсюдження або використання матеріалів, захищених авторським правом, без дозволу автора або з порушенням договору про використання таких матеріалів.

Такі дії є правопорушенням і переслідуються за законом. Однак, небажання деяких користувачів платити за працю автора призводить до того, що захищені авторським правом матеріали курсів стають публічно доступними необмеженому колу осіб.

Дана робота не призводить способи захисту інтернет-ресурсу від таких дій, але розглядає спосіб скоротити витрати через швидку локалізацію джерела витоку, а саме акаунта користувача, який став поширювати медіа-дані купленого курсу.

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ В ОБЛАСТІ ЦИФРОВОЇ СТЕГАНОГРАФІЇ

1.1 Області застосування стеганографії

На сьогоднішній день стеганографія використовується для захисту авторських прав, приховання зв'язку, автентифікації, для відстеження порушників (відбитків пальців), додавання додаткової інформації (наприклад, субтитрів до відео), додавання підписів до зображень, захист цілісності зображення (виявлення шахрайства), контроль копіювання при DVD-записі та в інтелектуальних браузерах, для автоматичного надання інформації в доступі та авторських правах, тощо (рис 1.1). Приклад використання прихованого зв’язку: організація резервного каналу, наприклад, для дипломатичних установ, розташованих на території іноземних держав [3]. В [4-5] представлена класифікація областей застосування стеганографії.

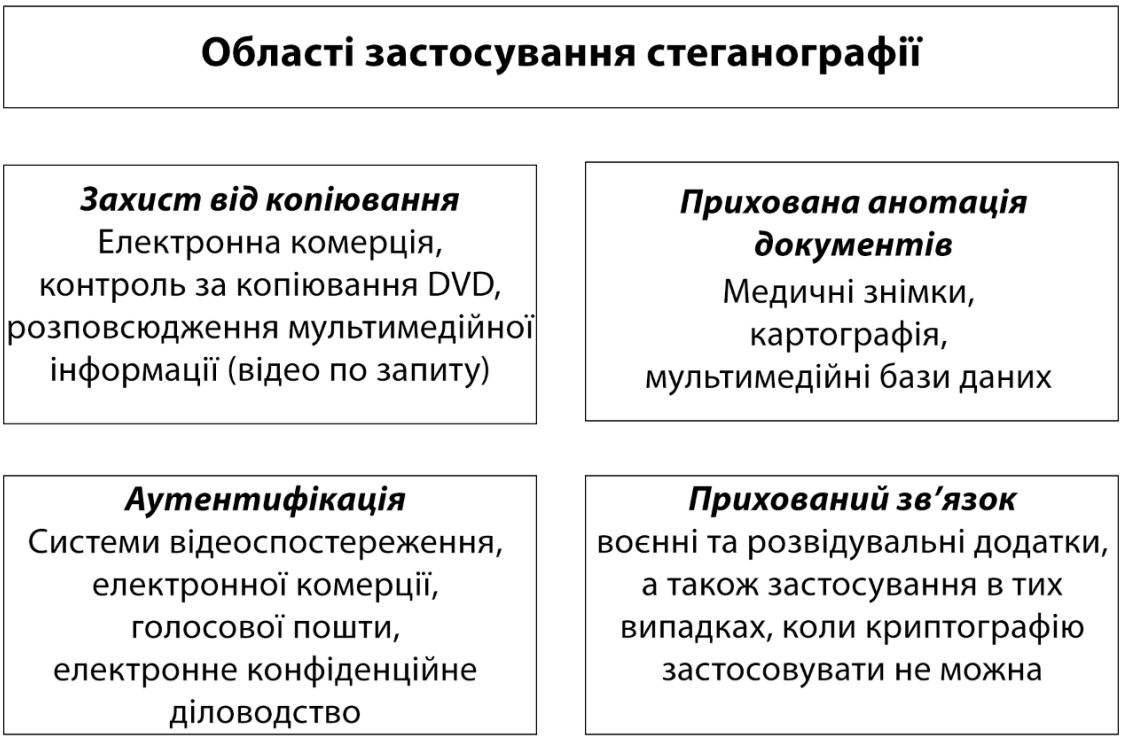

Рисунок 1.1 Класифікація областей застосування стеганографії

Стеганографія використовується для передачі прихованих повідомлень у країнах із суворою цензурою в Інтернеті. До речі, слід враховувати, що канали зв'язку можна контролювати (рис. 1.2). Пересилання зашифрованих повідомлень може викликати підозру і може призвести до обмеженого доступу до інфраструктури зв'язку. Тому в інтересах адресата є приховування наявності зв'язку. Ситуацію можна вирішити, використовуючи належний стеганографічний протокол [6].

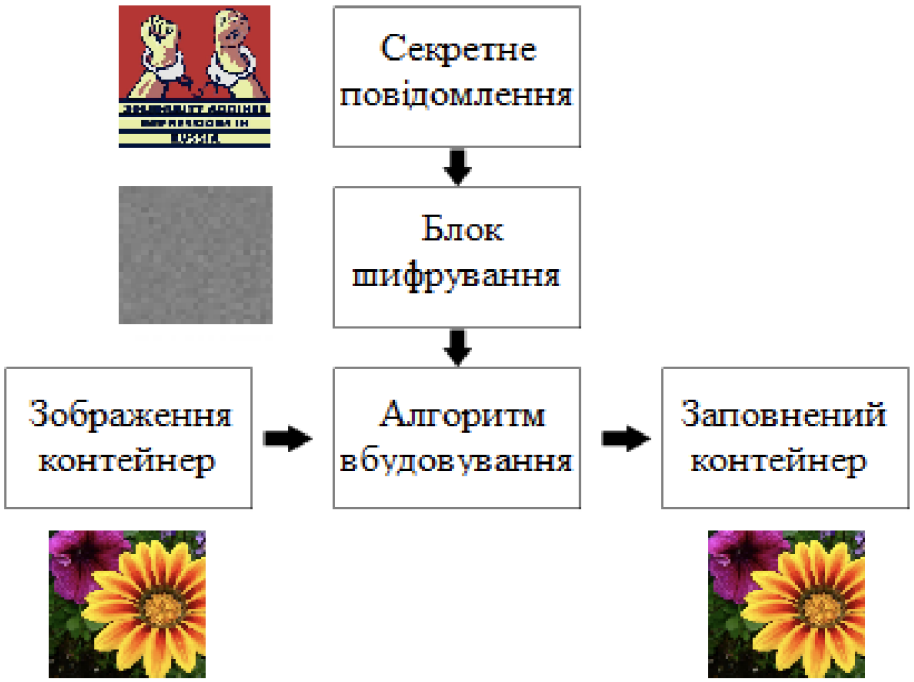

Рисунок 1.2 Блок-схема процесу вбудовування повідомлення при прихованому зв’язку

Завдяки швидкому розвитку мультимедійних технологій, постало питання про захист авторських прав на зображення, тобто аутентифікацію (рис. 1.3).

Прикладами можуть бути аудіо, фото та відеозаписи, тощо. Переваги, які дає представлення та передача інформації у цифровій формі можуть бути зведені до нуля, бо з нею можлива крадіжка або модифікація. Автор цифрового зображення може "підписати" зображення так, щоб неможливо було віднести авторство комусь іншому. Авторська інформація не може бути додана до файлу зображення, а також не може бути помітно надрукована на зображенні, тому що такі підписи можна

легко видалити або замінити. Криптографічні цифрові підписи використовуються лише для автентифікації каналу зв'язку, але не можуть захистити зображення, розміщене на веб-сторінці. Відмінним способом захисту буде введення в зображення невидимого, стійкого, захищеного ЦВЗ. При цьому автор зберігає оригінальне зображення. Для того, щоб довести своє авторство або навпаки розкрити підробку, автору необхідно перевірити наявність вбудованого ЦВЗ. Досвідчений злодій може спробувати видалити оригінальний водяний знак або вставити свій підпис на зображення. Але це не принесе йому успіху, тому що сліди обох ЦВЗ будуть присутні на зображенні, в той час, як автор може надати вихідний файл [6].

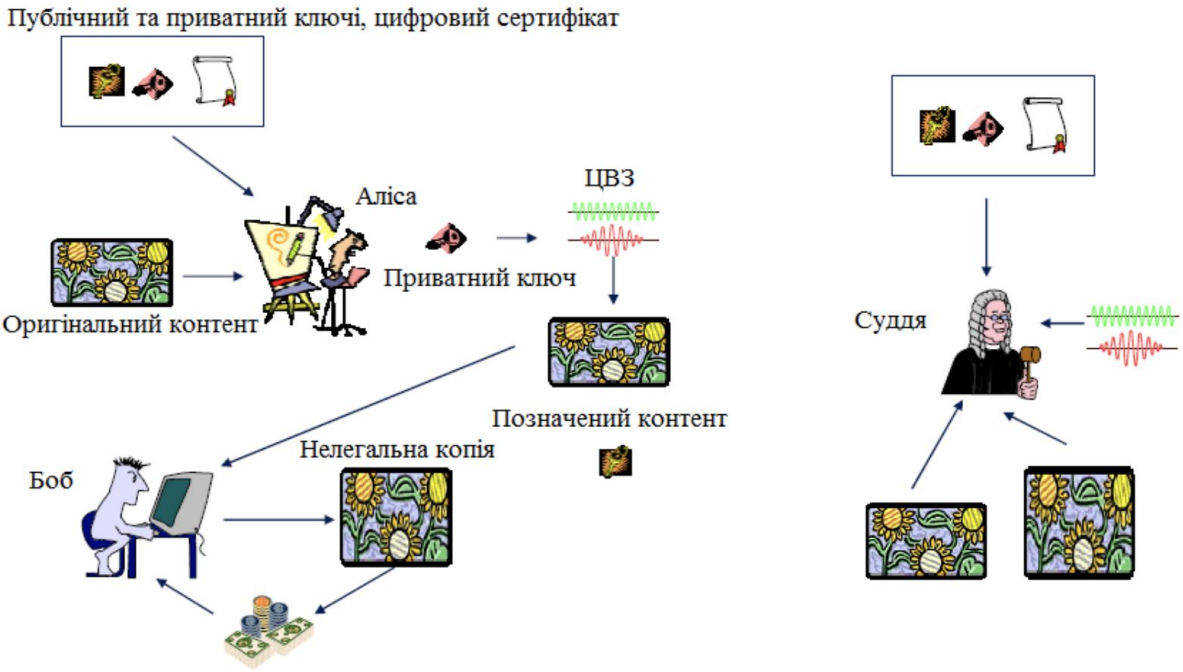

Рисунок 1.3 Блок-схема процесу вбудовування ЦВЗ з метою захисту авторських прав

Технологія введення ідентифікаційних номерів виробника має багато спільного з технологією ЦВЗ. Різниця в тому, що кожна копія захищена і має свій власний унікальний номер впровадження (звідси назва - дослівно «відбитки пальців"). Цей ідентифікаційний номер дозволяє виробнику відстежувати подальшу долю свого продукту [3]. На приклад, продаж ліцензійних дисків. Через унікальний

номер легко визначити, хто саме їх робить і поширює незаконні копії (рис. 1.4). Іншим сценарієм може бути поширення конфіденційної інформації (відео, фото) декільком депутатам і відстеження того, хто обманщик і через кого відбувається витік інформації ворогові. В цьому випадку неможливо використання видимого підпису, тому що таке маркування буде виглядати підозріло і може бути легко видалене. Ідентифікаційні знаки повинні бути невидимими і бути присутнім у кожному зображенні. Крім того, вбудовування має бути достатньо стійким, щоб не зруйнуватися при багаторазовому копіюванні та редагуванні.

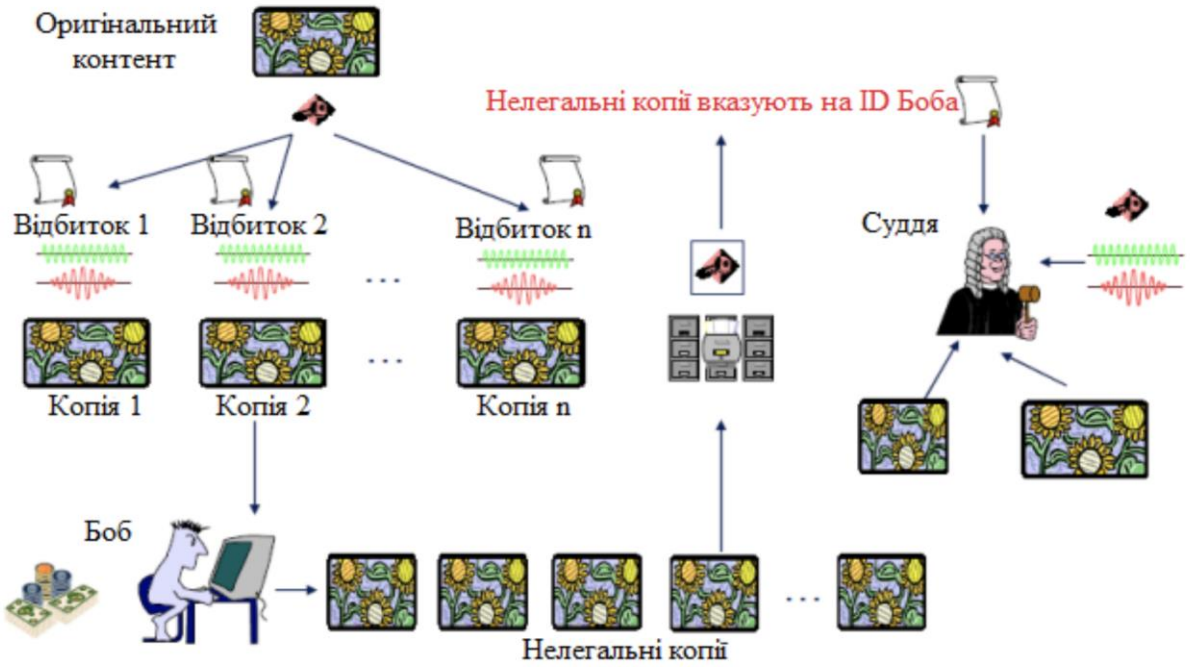

Рисунок 1.4 Блок-схема процесу вбудовування ідентифікаційних номерів з метою відстеження порушника

Прикладом вбудовування заголовків може бути: нанесення легенди на карту, підписи медичних знімків і т. і. Метою є зберігання різної інформації в одному цілому. Напевно, це єдиний додаток стеганографії, де в явному вигляді відсутній потенційний зловмисник. Типовими прикладами є субтитри, дубляж фільмів декількома мовами, відстеження використання даних (файл історії). Телевізор, Відеомагнітофон, DVD-програвач, і інші відео прилади можуть отримувати доступ і декодувати додатковий текст (субтитри) для кожного кадру режимі реального часу, і

зображати його на екрані телевізора. Хоча цей процес може бути реалізовано швидше шляхом додавання інформації, ніж непомітного вбудовування, однак вимоги до пропускної здатності і необхідність зміни формату іноді можуть не дозволити це зробити [6]. Однією зі сфер застосування стеганографії є захист цілісності зображення, що дозволяє виявляти випадки шахрайства (рис. 1.5). На жаль, в даний момент, цифрові зображення не можуть бути використані в суді як докази через легкість виготовлення цифрових підробок і складністю розкриття маніпуляцій з зображеннями. Вбудовування водяного знаку в цифрові зображення для того, що б виявити місця і ступеня зміни зображення буде відігравати важливу роль в виявленні цифрового шахрайства, і може бути використано в суді. Перевага використання ЦВЗ очевидна: ЦВЗ не залежать від формату зображення, не можуть бути видалені щоб уникнути підробок і не збільшують пропускну здатність (на відміну від додавання заголовків). Пристрій формування зображення, такий як відео камера, сканер або цифровий фотоапарат, маркує зображення унікальним, стійким, захищеним водяним знаком, перш ніж вони будуть відправлені для відображення на інший пристрій або ж збережені на електронному носії або, наприклад, комп'ютер [6].

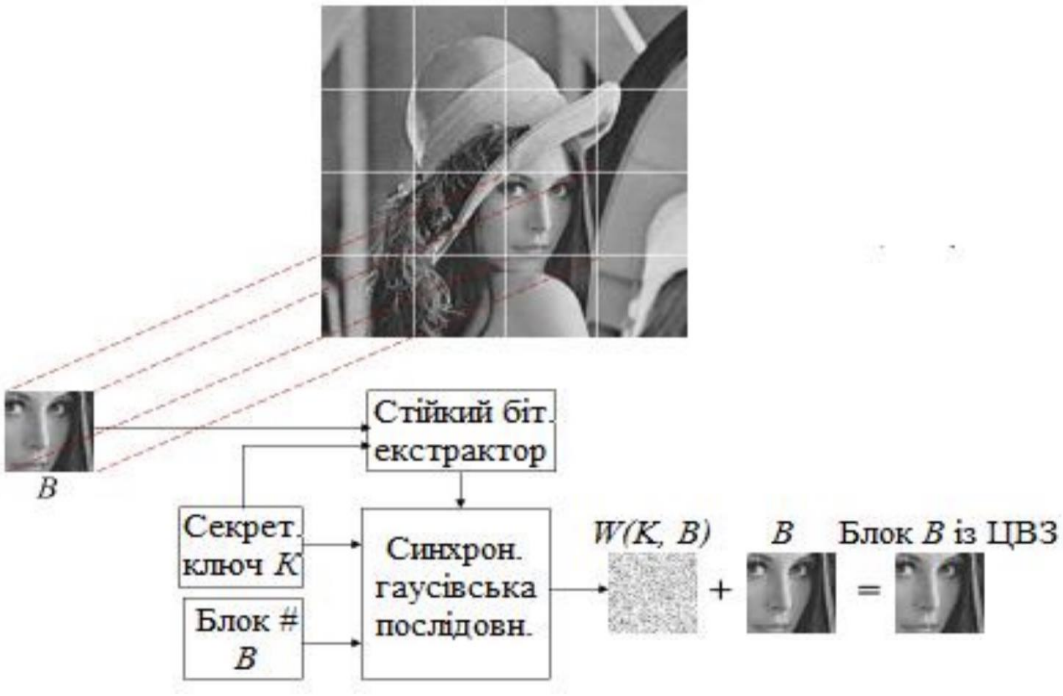

Рисунок 1.5 Блок-схема процесу вбудовування ЦВЗ для захисту цілісності зображення

Широкою сферою застосування стеганографії є управління копіюванням. Комерційно поширювані фільми часто мають стійкий, невидимий водяний знак, який показує чи можливе копіювання фільму [6]. Той же DVD-програвач може отримувати доступ до водяного знаку та відмовитися від подальшого копіювання фільму на інший диск (рис. 1.6).

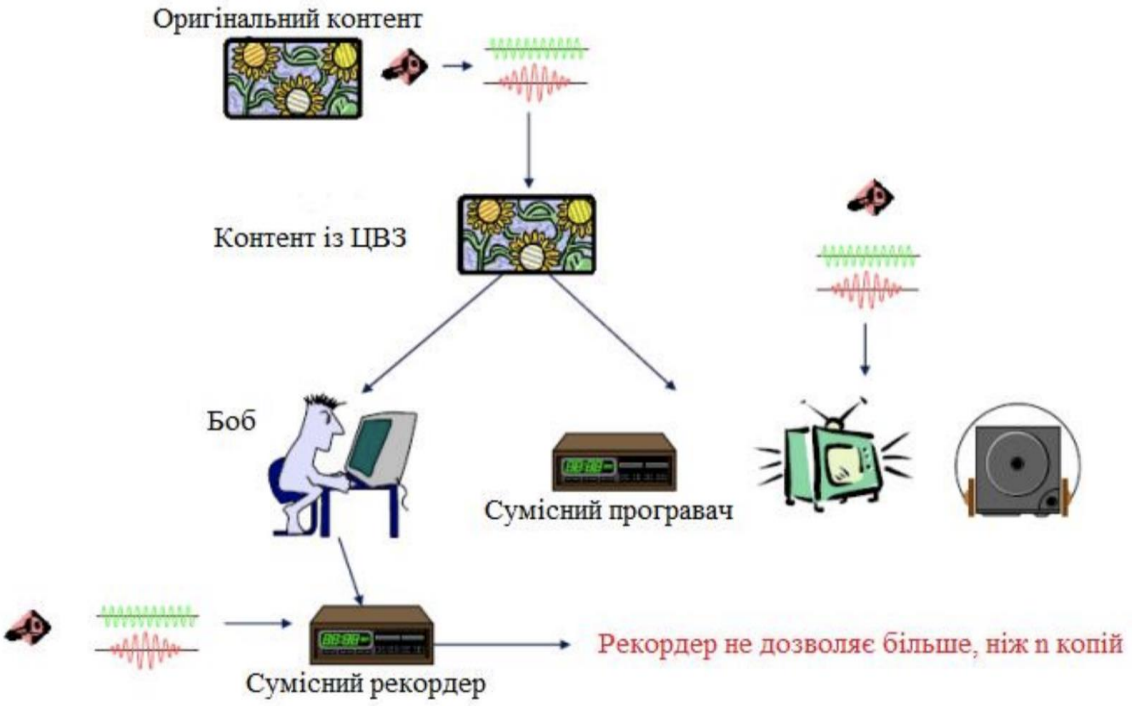

Рисунок 1.6 Блок-схема процесу вбудовування ЦВЗ для управління копіюванням при записі DVD

Методи стеганографії ще використовуються для автоматичного надання інформації про авторські права в інтелектуальних браузерах. Після того, як зображення загрузилось, воно перевіряється на наявність водяних знаків [6]. Якщо виявляються певні водяні знаки, то зображення не відображається і автоматично стирається з пам'яті комп'ютера. Іншим застосуванням є відображення авторської інформації кожного зображення, яким оброблялося зображення, додатками, такими як PhotoShop або Paint і т. п.

1.2 Математична модель стеганосистеми як системи передачі інформації

Узагальнена структурна схема стеганосистеми як системи передачі інформації наведена на рис. 1.7 [5].

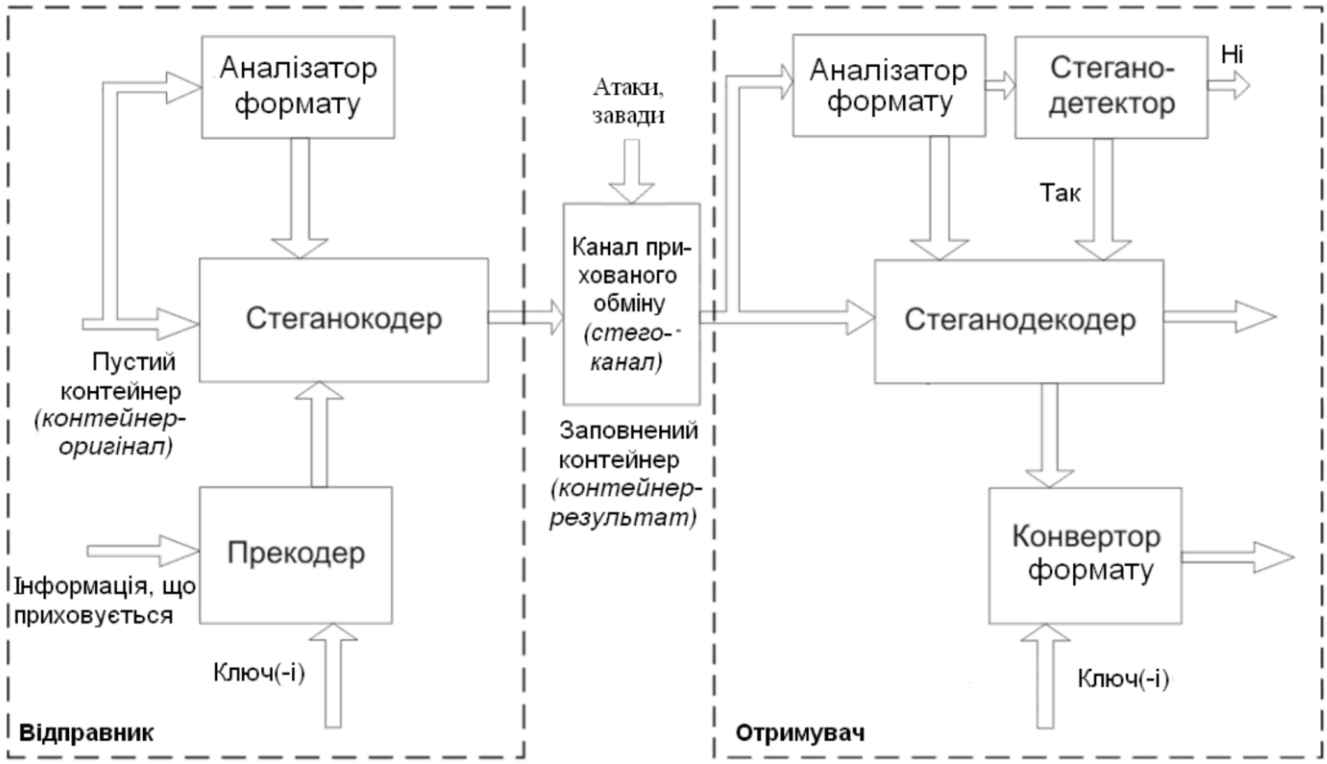

Рисунок 1.7 Структурна схема стеганосистеми як системи передачі інформації

Передача файлу від відправника до отримувача починається з обробки секретної інформації, яку необхідно приховати, – повідомлення. Початкова обробка відбувається за допомогою прекодера. Це необхідно для підвищення захищеності та стійкості стеганосистеми до спотворень. Найчастіше для цього використовується ключ, який зумовлює секретний алгоритм, що визначає порядок внесення повідомлення у контейнер [3].

Терміном контейнер називають несекретну інформацію, яка використовується для маскування повідомлення. Як повідомлення і контейнера можна використовувати не тільки звичайний текст, а й файли іншого формату, наприклад мультимедійного.

Порожній контейнер (контейнер оригінал) - це контейнер, який не містить прихованої інформації.

Заповнений контейнер (контейнер результат, стего) - контейнер, який вже містить сховану інформацію (наприклад, ЦВЗ). Основна вимога при цьому - контейнер-результат не повинен візуально відрізнятися від контейнера - оригіналу.

Розглядають два основних типи контейнерів: потоковий і фіксований.

Потоковий контейнер це послідовність бітів, він постійно змінюється. Повідомлення вставляється в нього в реальному масштабі часу, тому в заздалегідь не визначити, чи вистачить місця для всього повідомлення. Приклад потокового контейнера це аудіо чи відео запис.

Основною ж проблемою є виконання синхронізації, визначення початку і кінця послідовності. Якщо у данній послідовності існують біти синхронізації, заголовки пакетів і т.ін., то прихована інформація може міститися одразу ж після них. Складність організації синхронізації є перевагою з точки зору забезпечення прихованості передачі.

У фіксованому контейнері розміри і характеристики контейнера заздалегідь відомі. Прикладом фіксованого є статичне зображення, розміри якого і, як наслідок, потенціал для зберігання повідомлень нам заздалегідь відомі. Це позхволяет проводити всройку даних оптимальним способом. Тому розглянемо фіксований контейнер («контейнер»).

Контейнер може бути обраним, випадковим та нав’язаним. Обраний контейнер залежить від вбудованого повідомлення, а у крайньому випадку є його функцією. Такий тип контейнера найбільш характерний саме для стеганографії. Нав’язаний контейнер з’являється у випадку, коли той, хто надає контейнер, підозрює про ймовірність прихованої переписки і бажає запобігти їй. На практиці ж частіше за все мають справу із випадковими контейнерами.

Прихована інформація, в основному великого обсягу і пред'являє до контейнера суворі вимоги. Його розмір повинен як мінімум в рази перевищувати параметри даних, які будуть вбудовуватися. Чим розмір контейнера більше, тим простіше приховати інформацію.

Перед тим як виконати вкладення повідомлення контейнер, йому необхідно надати певного зручного виду. Крім цього, перед упаковкою в контейнер, для підвищення захищеності секретної інформації останню можна зашифрувати стійким криптографічним кодом. У багатьох випадках також бажано стійкість отриманого стеганоповідомлення до спотворень (в тому числі і зловмисним).

При передачі інформація, яка використовується як контейнер, може видозмінюватися (іноді використовуються алгоритми призводять до втрати даних): змененяются обсяг, перетворюється в інший формат тощо. Тому для кращого збереження вбудованого повідомлення можна використати код з виправленням помилок або завадостійке кодування.

Початкову обробку інформації, що ховається, виконує прекодер. В якості однієї з найважливіших попередніх обробок повідомлення (а також контейнера) можна назвати обчислення його узагальненого перетворення Фур'є. Що дозволяє здійснити вбудовування даних в спектральну область, що значно підвищує її стійкість до спотворень. Для підвищення секретності вбудовування первинна обробка за допомогою ключа.

Стеганокодер - пристрій для упаковки повідомлення в порожній контейнер, з огляду на формат цьоого контейнера.

Ключ - це елемент стеганосистеми, який використовується для вбудовування та вилучення повідомлень. Він параметризує алгоритм, який визначає порядок вбудовування повідомлення в контейнер. Ключ відомий тільки відправнику та одержувачу стеганоконтейнера.

Прихована інформація вбудовується згідно ключа в ті біти, зміна яких не призведе до суттєвих спотворень контейнера. Ці біти утворюють стеганографічний канал.

Стеганографічний канал - це канал передачі контейнера-результату. Коли контейнер з повідомленням знаходиться в стеганографічному каналі, то на нього можуть діяти як навмисні атаки, так і випадкові завади. У стеганодетекторі за форматом даних контейнера, визначається наявність в контейнері прихованих

даних. Ці зміни можуть з'явитися через помилки в каналі зв'язку, під час операції по обробці сигналу або навмисних атак хакерів.

Стеганодетектори діляться на ті, які призначені тільки для виявлення вбудованого повідомлення і пристрої, призначені для знаходження цього повідомлення в контейнері, - стеганодекодери [3].

У стеганосистемі об'єднані обидва типи інформації так, щоб вони по-різному сприймалися принципово різними детекторами. В якості одного з детекторів виступає система виділення прихованого повідомлення, в якості іншого - людина, яка пред'являє до системи передачі вимоги, і це досить важко формалізувати.

Для того, щоб стеганосистема була надійною і якісною, при її проектуванні необхідно виконати ряд вимог:

- Заповнений контейнер повинен візуально не відрізнятися від пустого. Для задоволення цієї вимоги слід, здавалося б, впроваджувати приховане повідомлення візуально незначні області сигналу. Однак, ці ж області використовують і алгоритми стиснення. Тому, якщо зображення надалі піддаватиметься стисненню, то приховане повідомлення може бути зруйнованим. Отже, біти повинні вбудовуватися в візуально значущі області, а непомітність може бути досягнута використанням спеціальних методів.

- Стеганосистема ЦВЗ повинна мати низьку ймовірність помилкового виявлення прихованого повідомлення в сигналі, який його насправді не містить. У деяких додатках таке виявлення може призвести до серйозних наслідків.

- Повинна забезпечуватися необхідна пропускна здатність.

- Стеганосистеми повинна мати прийнятну обчислювальну складність реалізації. При цьому можлива асиметрична за складністю реалізації система ЦВЗ, тобто складний стеганокодер і простий стеганодекодер.

1.3 Аналіз якісних характеристик стеганосистеми

Основне завдання будь-якої стегосистеми – вбудувати повідомлення в контейнер так, щоб сторонній спостерігач не міг помітити різницю між

оригінальним та модифікованим контейнерами. Зазвичай система будується так щоб забезпечити певний компроміс її базових характеристик, до яких відносяться непомітність, стійкість, захищеність, пропускна здатність утвореного стеганоканалу та обчислювальна складність реалізації.

Аналіз існуючої літератури [3-5,7], показав, що немає чіткого розмежування між поняттями стійкості (robustness) та безпеки (security). Часто ці поняття вживаються взаємозамінно. Перша спроба розділити дані поняття була зроблена в [7] та далі розвинена у роботах [8-9]. Аналіз показав, що безпека – це базова характеристика, основним завданням якої є захист від умисник атак порушника, а робастність повинна забезпечувати захист від інших видів атак. Підкреслимо, що поняття стійкості не включає в себе атаки на методи вбудовування, що ґрунтуються на знаннях алгоритму вбудовування або вилучення. Під стійкістю мається на увазі, стійкість до нецільових модифікацій, або узагальнені операції з зображеннями.

Термін "робастність" (robustness - англ.) утворений від robust - міцний, грубий (англ.). Порівняйте з назвою одного із сортів кави - robusta. Мається на увазі, що робастні процедури повинні "витримувати" помилки, які тими чи іншими способами можуть потрапляти в вихідні дані або спотворювати передумови використовуваних ймовірносно-статистичних моделей.

Термін "робастний" став популярним в нашій країні в 1970-і роки. Спочатку він використовувався фактично як звуження терміна "стійкий" на алгоритми статистичного аналізу даних класичного типу (не включаючи теорію вимірювань, статистику нечислових та інтервальних даних). Потім реальна сфера його застосування звузилася.

Робастність - це здатність прихованого повідомлення залишатися неушкодженим, навіть якщо стего-медіа піддають трансформації, лінійній та нелінійній фільтрації, масштабуванню, розмиванню, обрізанню та іншим атакам [10]. Іншими словами - це стійкість ЦВЗ до різного роду перешкод і спотворень. Саме робастним ЦВЗ присвячено більшість досліджень.

ЦВЗ можуть бути трьох типів: робастні, крихкі й напівкрихкі (semifragile) [3].

Крихкі водяні знаки знищуються при невеликій зміні заповненого контейнера. Їх використовують для розпізнавання сигналів. Для захисту медіа інформації це надзвичайно важливо, оскільки законний користувач або, навіть, сам автор може захотіти змінити дані, наприклад стиснути зображення. Також відзначимо, що крихкі ЦВЗ мають не тільки вказати на факт модифікації контейнера, але й також вигляд та розташування цієї модифікації.

Напівкрихкі ЦВЗ є стійкими до одних атак і нестійкими до інших. Загально кажучи, всі ЦВЗ можна віднести до цього типу. Проте напівкрихкі ЦВЗ навмисно проектуються так, щоб бути нестійкими відносно певних операцій. Наприклад, вони допомагають стискати зображення, але забороняють вирізати або вставляти в нього сторонній фрагмент.

Невидимість (невідчутність, imperceptibility) – характеристика, що відповідає за нездатність людським зором без використання спеціальних засобів виявити приховане повідомлення. Це поняття спирається суто на властивості зорової системи людини (ЗСЛ). Скрита інформація непомітна, якщо людина не може відрізнити носій з прихованою інформацією від носія без неї. Загальноприйнята схема експерименту, яку часто називають "сліпий тест", заснована на тому, що суб’єктам пропонують в довільному порядку серед великої кількості носіїв з та без вбудованої інформації обрати, які саме носії містять приховані дані [6].

На практиці кількісними показниками невідчутності є ввідношення сигнал/шум SNR, середньоквадратична похибка MSE, максимальна різниця MD та інші [5].

Система захищена, якщо вбудована інформація не може бути видалена атаками, спрямованими на приховані дані та заснованими на завчасно відомому алгоритмі вбудовування-вилучення (крім секретного ключа), й знанні хоча б одного носія з прихованим повідомленням. Захищеність також містить в собі процедурні атаки, наприклад атаки IBM [11] або атаки на ґрунтуючись на знаннях про часткові зміни носія на основі наявності вкладення [12].

Складність вбудовування і вилучення – кількість операцій та дій, що мають бути виконані для вбудовування і викриття прихованого повідомлення.

Стеганосистема повинна мати прийнятну обчислювальну складність реалізації. До того ж реалізація стеганографічної системи передачі інформації може бути асиметричною за складністю, наприклад складний стеганокодер і простий стеганодекодер і навпаки.

Перераховані вимоги конкурують одна з одною і не можуть бути оптимальними разом. Якщо необхідно приховати велике повідомлення в зображенні, то неможливо вимагати повної невидимості й хорошої стійкості. Необхідно знайти компроміс. З іншого боку, якщо потребується стійкість до великих спотворень, то повідомлення, що має бути надійно сховане, не може бути дуже довгим. Це ми можемо бачити на рисунку на рис. 1.8.

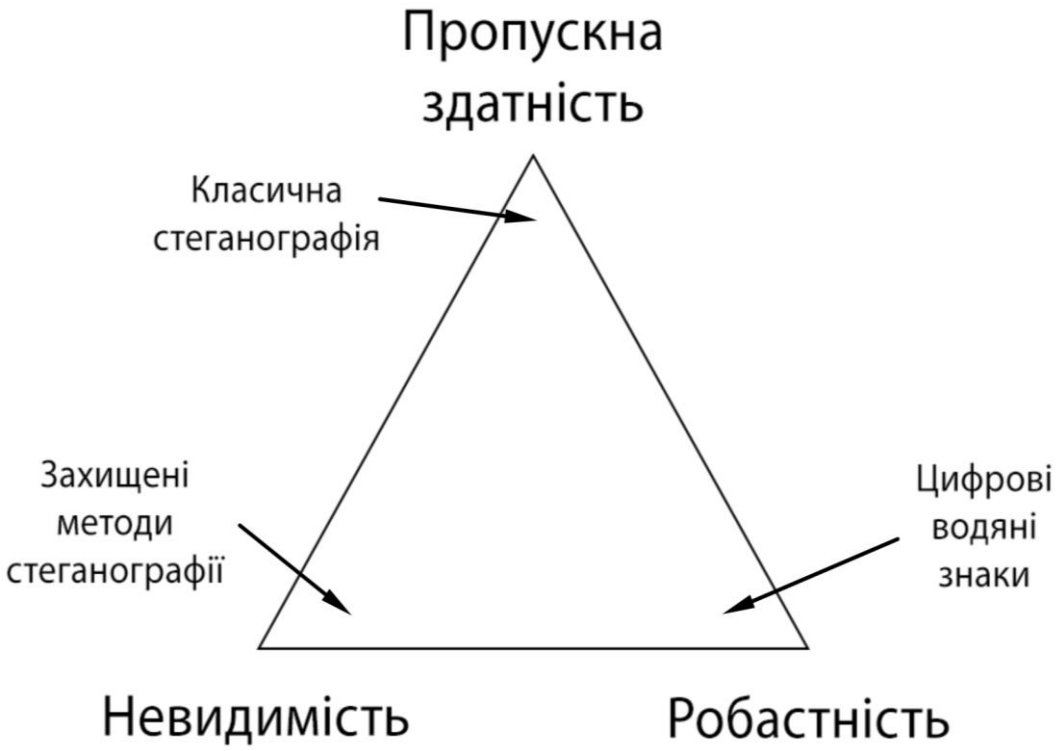

Рисунок 1.8 «Магічний трикутник» ключових характеристик стеганосистем

1.4 Висновки до розділу 1

Стеганографія використовується для захисту авторських прав (контроль розповсюдження медіаданих, електронна комерція), для прихованої анотації документів (медичні знімки, картографія, мультимедійні бази даних), для

аутентифікації (голосова пошта, доступ до даних, систем відеоспостереження тощо). За допомогою стеганографії можна організувати прихований зв’язок (воєнні та розвідувальні додатки, в країнах з інтернет-цензурою тощо).

Були розглянуті основні положення комп’ютерної та цифрової стеганографії. Побудовано узагальнену структурну схему стеганосистеми як системи передачі та розглянуто основну термінологію стеганографічних систем.

Методи комп’ютерної стеганографії повинні відповідати основній вимозі - бути непомітним, щоб сторонній спостерігач не міг знайти різницю між оригінальним та модифікованим контейнерами. Також необхідно забезпечити певний компроміс якісних характеристик системи, до яких належать непомітність, стійкість (робастність), захищеність, пропускна здатність утвореного стеганоканалу та обчислювальна складність реалізації.

РОЗДІЛ 2. РОЗГЛЯД МЕХАНІЗМІВ ВНЕСЕННЯ ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ. ВНЕСЕННЯ ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ В ЗОБРАЖЕННЯ, ЗВУКОВІ ФАЙЛИ ТА ВІДЕО

2.1 Узагальнена класифікація стеганографічних методів

Сучасні стеганографічні методи розвиваються у двох основних напрямках:

1) методи, засновані на використанні спеціальних властивостей комп'ютерних форматів;

2) методи, засновані на надмірності аудіо і візуальної інформації.

Більшість методів опираються на два принципи:

1) файли, що не потребують абсолютної точності (файли з зображенням, звуком тощо), вони можуть бути модифіковані (до певного ступеня) без втрати основоної функціональності;

2) сенсорна система людини не здатна точно розрізняти несуттєві зміни у модифікованих стеганографічним способом файлах та/або відсутній інструментарій, який міг би виконати такий тип задачі.

Методи цифрової та комп'ютерної стеганографії в загальному вигляді можуть бути класифіковані, спираючись на відомі публікації [3, 5, 13-15] та вибираючи той чи інший класифікаційний критерій (рис. 2.1).

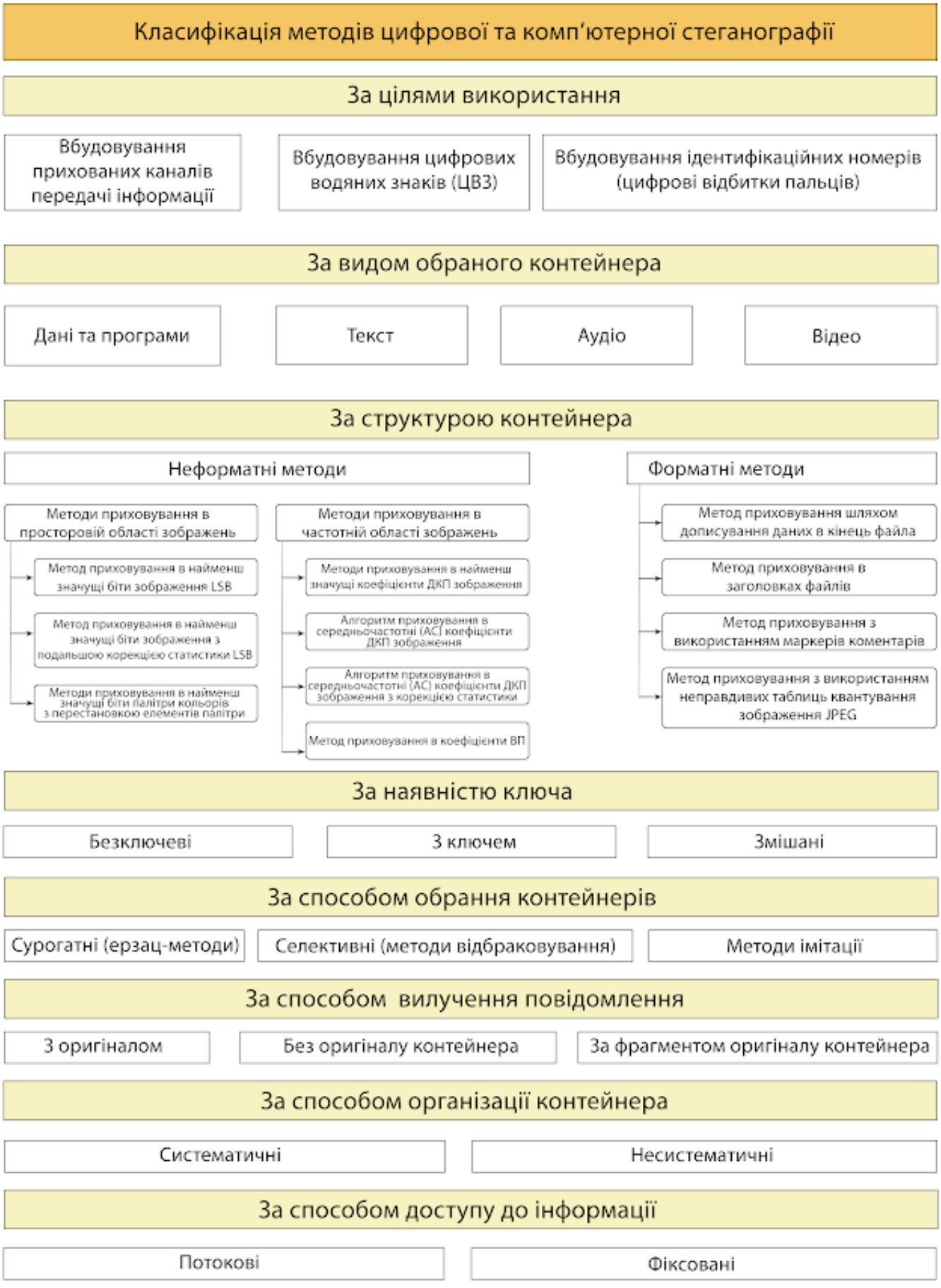

Рисунок 2.1 Класифікація методів цифрової та комп'ютерної стеганографії

За цілями використання методів цифрової та комп'ютерної стеганографії загальновизнаними є три напрямки:

- вбудовування прихованих каналів передачі інформації - метою вбудовування є приховування факту передачі інформації;

- вбудовування цифрових водяних знаків (ЦВЗ) - мета вбудовування складається в підтвердженні справжності зраджувати даних і в запобіганні несанкціонованого доступу до них;

- вбудовування ідентифікаційних номерів (цифрові відбитки пальців) - з метою прихованої анотації й аутентифікації переданої інформації.

Найбільш популярним напрямком, який отримав розвиток останнім часом, є вбудовування ЦВЗ. Це багато в чому визначається необхідністю забезпечення захисту від несанкціонованого поширення інформації, що є інтелектуальною власністю [17-19]. Організація стеганографічних каналів є більш актуальним напрямком для організацій і відомств, в яких забезпечення безпеки інформації є пріоритетним вимогою. У зв'язку з тим, що цілі, переслідувані при встановленні ЦВЗ і при організації каналів різні, то і основні вимоги, що пред'являються до розроблюваних стегометодам мають ряд відмінностей. Оскільки ЦВЗ призначений для захисту від несанкціонованого копіювання, то знання порушником про вбудовуванні ЦВЗ в об'єкт, що захищається не є критичним, на відміну від робастності ЦВЗ, оскільки основний атакою, яка застосовується в даній ситуації є геометрична. При цьому завдання підвищення прихованої пропускної здатності не варто, як така, на відміну від необхідності забезпечення високої достовірності прийому біт ЦВЗ. При організації прихованого каналу же скритність вбудовування є визначальним вимогою і основний атакою є візуальна атака і статистичний аналіз [20-22]. Крім того, оскільки при організації стегоканалів мова йде про передачу інформації, то необхідно забезпечити необхідну приховану пропускну здатність і достовірність прийому приховуваних даних не гірше мінімальної залежності від виду переданих повідомлень.

За видом контейнера, обраного для вбудовування стеганографічні методи класифікують [3, 5, 23] на методи, що піддають модифікації дані і програми, текст,

аудіо та відео. У зв'язку з тим, що організація прихованих вкладень можлива переважно завдяки надмірності того виду даних, яка була обрана носієм, то очевидна популярність застосування для цього завдання аудіо і відео даних, як найбільш надлишкових. Стегометоди організації прихованих каналів використовують в якості контейнера в основному аудіо та відеоданих.

Відповідно до того, яка область в структурі контейнера підлягає модифікації, розрізняють форматні та неформатні стеганографічні методи. Застосування форматних обмежене невисокою стеганографічною стійкістю (робастністю) при досить низькій пропускній здатності і більш застосовно для організації ЦВЗ. Другий напрямок є більш перспективним і базується на модифікації параметрів простору приховування файлу, що характеризують безпосередньо дані самого зображення або звуку. У цій області розроблені і добре апробовані стійкі до виявлення стеганографічні алгоритми, що забезпечують достатню місткість контейнерів для вкладення приховуваних повідомлень або програм. Зокрема, це і офіційно визнані алгоритми стеганографії F5 і OutGess. Оскільки прихована пропускна здатність безпосередньо залежить від надмірності контейнера, то найбільш застосовними в інтересах організації прихованої передачі інформації є рухомі і нерухомі зображення.

Процес вибору контейнера, вбудовування прихованої інформації, подання цих процесів у вигляді моделей в загальному вигляді описані в роботах [3, 5]. Часто використовують наступний принцип вбудовування даних. Практично будь-який контейнер в результаті обробки може бути представлений послідовністю з N біт. Процес приховування інформації починається з визначення біт контейнера, які можна змінювати бітами вбудовуваної послідовності без внесення помітних спотворень. Одним з перших методів вбудовування стего-вкладення є метод, заснований на заміні найменш значущого біта контейнера (НЗБ). Цей метод простий у реалізації і дозволяє досягти максимуму прихованої пропускної здатності, однак має найменшу скритність та робастність. Це й сприяло подальшому вдосконаленню методу НЗБ, а також пошуку нових методів і супроводжувалося ускладненням алгоритмів вбудовування та вилучення, а також зниженням прихованої пропускної

здатності. У роботах [24-26] зазначено, що вбудовування в просторову область зображень, характеризується роздвоєнням піку гістограми, який є демаскуючою ознакою, що визначає доцільність проведення модифікації оцифрованих спектральних складових контейнера.

За наявністю ключа стеганографічні методи ділять на три групи: безключеві, з ключем та гібридні (змішані) [27].

Для функціонування стеганосистеми без ключа, окріма алгоритму графічного перетворення, інші додаткових дані, на зразок стего-ключа, не потрібні.

Таким чином, захищеність безключевої стеганосистеми базується лише на секретності стеганографічних перетворень, які використовуються.

Ключова стеганосистема в свою чергу поділяється на підсистеми з відкритим та закритим ключем. Стегосистема з відкритим ключем повинна мати закритий канал зв’язку для передачі стегоключа і повинна забезпечувати вищий рівень захищеності повідомлення, ніж система без ключа. Також така система потребує більше затрат на передачу стегоключа. Стеганосистема з відкритим ключем працює аналогічно з криптографічними алгоритмами, проте необхідно зазначити, що стеганоключ приховує місце вбудовування даних в контейнері, а не шифрує їх.

Гібридні стегосистеми можуть використовувати і відкритий, і секретний ключ.

Відповідно до способу вибору контейнера виділяють сурогатні, імітаційні та селективні стеганографічні методи [27]. В сурогатних (або безальтернативних) методах можливість вибору контейнера відсутня і для приховання повідомлення обирають перший контейнер, який не є оптимальним у більшості випадків(ерзац-контейнер). Селективні методи передбачають відтворення спеціальних статистичних характеристик шуму контейнера прихованим повідомленням. Для цього необхідно генерувати багато альтернативних контейнерів, з подальшим обраннямна й оптимальнішого з них (шляхом відбраковування) для конкретного повідомлення. Для такого підходу окремим випадком є обчислення хеш-функції кожного контейнера. Після цього для приховування повідомлення обирають той контейнер, в якого хеш-функція співпадає зі значенням хеш-функції повідомлення(себто обраний контейнер є стеганограмою). В імітаційних

стеганографічних методах сама стеганосистема генерує контейнер. Існують кілька варіантів реалізації. Для прикладу, шум контейнера може бути зімітованим повідомленням, що приховується. Це можна реалізувати за допомогою спеціальних процедур, де приховане повідомлення кодується як шум й зберігається модель шуму. Прикладом такої реалізації є метод у програмі MandelSteg [28], яка генерує фрактал Мандельброта в якості контейнера.

За способом організації контейнери можуть бути розділені на систематичні й несистематичні [27]. У перших можна вказати конкретні місця стеганограми, де знаходяться інформаційні біти власне контейнера, а де шумові біти, призначені для приховування інформації (як, наприклад, у найпоширенішому й найпростішому методі найменшого значущого біту). Для несистематичної організації контейнера такий поділ не можливий. У цьому разі для виділення прихованої інформації необхідно обробляти вміст всієї стеганограми.

За способом доступу до приховуваної інформації розрізняють методи потокових (безперервних) і фіксованих (обмеженої довжини) контейнерів [27].

За принципом приховування методи комп’ютерної стеганографії поділяють на: методи з безпосередньою заміною та спектральні методи [27]. Якщо перші, за допомогою надлишковості інформаційного середовища в просторовій (для зображення) або часовій області (для звуку), використовують заміну малозначимої частини контейнера бітами секретного повідомлення, то другі для приховування даних використовують спектральне представлення елементів середовища, куди вносяться дані для приховування (наприклад, до різних коефіцієнтів ДКП, перетворень Фур’є, Адамара, Хаара тощо). Саме використання властивостей надлишковості контейнера-оригінала є основним напрямком комп'ютерної стеганографії. Але при цьому треба брати до уваги те, що приховування інформації веде до спотворень деяких статистичних властивостей контейнера або ж порушення його структури. В окрему групу можна віднести методи, що використовують спеціальні властивості форматів представлення файлів [16]:

— спеціальне форматування даних (зсув слів, абзаців, речень, обирання визначених позицій літер);

— зарезервовані для розширення поля файлів, які зазвичай заповнені нулями й не рахуються програмою;

— використання незадіяних ділянок магнітних та оптичних носіїв;

— видалення заголовків-ідентифікаторів тощо.

Для таких методів характерний низький рівень прихованості, низька пропускна здатність й продуктивність.

Формалізація деяких відомих методів, які є основою для розробки нових підходів до організації стеговкладень, представлена в [26, 29] і зведена в таблицю 2.1.

Таблиця 2.1. Найбільш відомі підходи до організації прихованих каналів і ЦВЗ в відеоданих

Класифікація методів організації прихованих каналів в відеоданих			Правило вбудовування ρ	Пропускна здатність прихованого каналу $C_{стег}$	Скрит-ність U	Достовірність приховуваних даних $p_{пом}$	Число прихованих каналів
1. Напрямок цифрової стеганографії - вбудовування цифрових водяних знаків (ЦВЗ) з метою підтвердження даних							
2. Напрямок цифрової стеганографії - організація прихованого каналу з метою приховування факту передачі							
В просторовій області	Найменш значущого біта		$\rho^1 = I + S$	$C^1_{стег} = \max$	$U^1 = \min$	$p_{ош} = 10^{-3}$, без стиснення	1
	К.Ю. Цвєтков, М.В. Коровін	К. Матсуі, К. Танака, С. Осборн, Дж. Фрідріх					
	Псевдовипадкового інтервалу (Д. Рамкуман, Дж. Симонс, В. Волошиновський)		$\rho^2 = I + S\|I_h = 1$	$C^2_{стег} < C^1_{стег}$	$U^2 > U^1$	$p_{ош} = 10^{-3}$, без стиснення	1
	Псевдовипадкової перестановки		$\rho^3 = I + S\|h = \varphi_i$, φ_i – ПВП	$C^3_{стег} < C^2_{стег}$	$U^3 > U^2$	$p_{ош} = 10^{-3}$, без стиснення	1
	В.І. Коржик	С. Моллег, А. Фіцман, І. Стиран					
	Блочного приховування (В.О. Хорошко, О.Д. Азаров, К.Ю. Цвєтков, М.В. Коровін)		$\rho^4 = \left(\frac{I}{D}\right) + S$, D - число блоків	$C^4_{стег} \approx C^3_{стег}$	$U^4 > U^3$	$p_{ош} = 10^{-3}$, без стиснення	1
В частотній області	Найменш значущого біта		$\rho^5 = I^{ДКП} + S$	$C^5_{стег} = \max$	$U^5 > U^1$	$p_{ош} = 10^{-3}$, без стиснення	1
	К.Ю. Цвєтков, В.Г. Грибунін, М.В. Коровін	Е. Кох					
	Відносної заміни		$\rho^6 = \begin{cases} S_0, \left(\frac{I^{ДКП}}{D}\right)_1 - \left(\frac{I^{ДКП}}{D}\right)_2 > P \\ S_1, \left(\frac{I^{ДКП}}{D}\right)_1 - \left(\frac{I^{ДКП}}{D}\right)_2 < P \end{cases}$, P - поріг	$C^6_{стег} < C^4_{стег}$	$U^6 \approx U^4$	$p_{ош} = 10^{-3}$, в процесі стиснення	1
	В.І. Коржик	Е. Кох, Дж. Жао					
	Бенгама-Меммона (Д. Бенгман, Н. Меммон, Б-Л Ео, М. Юнг)		$\rho^7 = \begin{cases} S_0, \begin{cases} \left(\frac{I^{ДКП}}{D}\right)_3 < \left(\frac{I^{ДКП}}{D}\right)_1 \\ \left(\frac{I^{ДКП}}{D}\right)_3 < \left(\frac{I^{ДКП}}{D}\right)_2 \end{cases} \\ S_1, \begin{cases} \left(\frac{I^{ДКП}}{D}\right)_3 > \left(\frac{I^{ДКП}}{D}\right)_1 \\ \left(\frac{I^{ДКП}}{D}\right)_3 > \left(\frac{I^{ДКП}}{D}\right)_2 \end{cases} \end{cases}$,	$C^7_{стег} < C^6_{стег}$	$U^7 > U^6$	$p_{ош} = 10^{-3}$, в процесі стиснення	1
	Фрідріх		$\rho^8 = (I)^{ДКП} + S\|(I')^{ДКП} < (I)^{ДКП} + 10^2\alpha$, I' – модифіковані відеодані, α – порогова функція	$C^8_{стег} < C^7_{стег}$	$U^8 \approx U^7$	$p_{ош} = 10^{-3}$, в процесі стиснення	1
	Е.А. Небаєва	Дж. Фрідріх					
	Розширення спектру		$\rho^{10} = (I)^{ДКП} + S \cdot \varphi_i$, φ_i – ПВП	$C^8_{стег} < C^{10}_{стег} < C^1_{стег}$	$U^{10} \approx U^9$	$p_{ош} = 10^{-3}$, в процесі стиснення	1
	К.Ю. Цвєтков, А.Е. Коревих, М.В. Коровін	Дж.-Р. Сміт, В.О. Комісі					
Статистичний І. Патіс, В.О. Хорошко, В.Г. Грибунін, М.Е. Шелест			$\rho^9 = \left[(I)_1^{ДКП}\right]^{\frac{E}{2}} \cup \left[(I)_2^{ДКП}\right]^{\frac{E}{2}}$, E – енергія зображення	$C^9_{стег} < C^8_{стег}$	$U^9 > U^8$	$p_{ош} = 10^{-3}$, в процесі стиснення	1

Розвиток методів вбудовування як в просторовій, так і в частотній області йшов, як правило, шляхом ускладнення правила вбудовування та пошуку функцій

для вибору біт, що підлягають заміні, максимально схожим на випадкову величину. В роботі [3] в якості альтернативних методів заміни вказані наступні.

Вбудовування шляхом інверсії біта: «1» може відповідати заміна 0> 1, «0» - заміна 1> 0. Вбудовування шляхом вставки біта безпосередньо перед бітом, що підлягає модифікації. При цьому значення біта ЦВЗ повинне бути протилежним значенню біта контейнера.

Вбудовування видаленням біта: для цього вибирають пари бітів «01» або «10», щоб вони відповідали різним значенням біта ЦВЗ. Потім перший біт пари видаляється.

Вбудовування з використанням біта-прапора: черговий біт контейнера (незмінний) є бітом ЦВЗ і вказує на інверсію попереднього біта-прапора.

Вбудовування з використанням граничних біт: відбувається аналогічно до вбудовування біт-прапора, але одному біту ЦВЗ відповідає не один, а кілька біт, що йдуть слідом за біт-прапором (непарне число). Якщо серед набору біт більше одиниць, то біт ЦВЗ дорівнює «1».

Вбудовування з використанням табличних значень. Для визначення біта ЦВЗ в попередньому методі, фактично, використовувалася перевірка на парність. Так само можна використовувати й будь-яке інше відображення безлічі біт в 1 біт, або знаходити його значення по таблиці. Можливе використання динамічно змінною таблиці, коли таблиця змінюється на кожному кроці або вибір значення з таблиці здійснюється псевдовипадково. Оскільки табличні значення (біти контейнера) знає і кодер, і декодер, то їх можна не передавати (непряма динамічна таблиця).

Вбудовування із застосуванням функції, яка оцінює статистику зображення і кореляційні зв'язки між елементами зображення, і подальше застосування цієї функції для кожного елемента зображення для визначення стегошляху. При цьому в якості опції може бути використана псевдовипадкова послідовність (ПВП).

Іншим підходом є вбудовування додаткової інформації за рахунок енергетичної різниці між коефіцієнтами контейнера, що характеризується малою зміною статистики зображення.

Окремою класифікаційною групою стегометодів є організація ЦВЗ і стегоканалів з використанням широкосмугових сигналів (ШСС) [20, 24, 30-32], ці методи ще називають методами з розширенням спектра. При цьому модифікації підлягає повністю все зображення, на відміну від вбудовування в частотну чи просторову область, де заміні підлягають лише ті елементи й коефіцієнти, до яких найменш чутлива система зору людини і зміна статистики зображення.

2.2 Огляд основних методів цифрової стеганографії

Для вибору відповідного методу для подальшої реалізації алгоритму проведемо порівняльний аналіз існуючих методів цифрової стеганографії.

Серед методів нанесення ЦВЗ можна відзначити просторові і частотні. Перші виконують запис знака безпосередньо на зображення, змінюючи значення кольоровості і інтенсивності окремих пікселів або блоків пікселів. Частотні методи працюють в спектральній області зображення.

2.3.1 Least Significant Bit - найменш значущий біт (НЗБ)

Просторовий метод, суть якого полягає в тому, що вбудовування секретної інформації проводиться в найменш значущі біти пікселя - як правило останній один або два. Таким чином, інтенсивність кольору даного пікселя змінюється незначно - досить для того, щоб записати фрагмент секретної інформації і зберегти загальний вигляд зображення дуже близьким до оригіналу.

Кодування зображень для веб-ресурсів проводиться за моделлю RGB, тобто кожен піксель являє собою 3-компонентну структуру, що описує значення інтенсивності для кожного з трьох базових кольорів - червоний, зелений і синій. [33] При цьому значення інтенсивності кожного компонента кодується 1 байтом (8 біт), це значення від 0 до 255 включно.

Приклади кодування пікселя:

- #000000 - білий піксель;

- #FFFFFF - чорний піксель;

- #FF0000 - червоний піксель;

- #00FFFF - жовтий піксель (максимальна інтенсивність зеленого і синього каналів).

Найчастіше для методу НЗБ вибирають синій канал тому до зміни інтенсивності цього кольору очей людини найменш сприйнятливий. Звичайно, можна використовувати і монохромні зображення, користуючись єдиним представленим каналом.

Для запису повідомлення в контейнериспочатку визначають **стегошлях** - масив пікселів, які будуть модифіковані, а потім модифікують останній біт пікселів в стегошляху відповідно до чергового кодованого біту повідомлення.

Таким чином, якщо ми кодуємо 3-й біт із повідомлення 0110010 в синій канал пікселя #FAFAFA, то значення синього каналу буде модифіковано таким чином:

- поточне значення: FA;

- бінарне представлення: 1111 1010;

- запис 3-го біта, тобто «1» в молодший значущий біт: 1111 1011.

Результуюче значення кольору пікселя: #FAFAFB

Такий метод відмінно підходить у випадках, коли заздалегідь відомо, що стего не буде зазнавати змін, тому що він неробастний (крихкий). При цьому можна записати дуже велику кількість інформації - буквально ще одне зображення всередину вихідного зображення.

В цьому випадку ЦВЗ є крихким і не підлягає відновленню навіть при невеликих змінах, що не підходить для нашої ситуації. Це ні в якому разі не означає, що даний алгоритм не підходить для цифрової стеганографії взагалі і не може ніде використовуватися. Навпаки, іноді до системи пред'являються вимоги, згідно з якими потрібно підтверджувати оригінальність контенту. Це можливо шляхом вбудовування водяного знака, який «розпадеться» при найменшому впливі на нього.

2.3.2 Дискретне косинусне перетворення

Суть методу дискретного косинусного перетворення (ДКП) концентрує енергію в області низьких частот, і так як людське око менш чутливе до високочастотних коливань, то високочастотні компоненти можуть бути оцифровані більш грубо. ДКП використовує метод диференційного вбудовування енергії, що активно застосовується при стисненні зображень алгоритмом JPEG і відео алгоритмом MPEG.

Алгоритм диференційного вбудовування енергії вибирає n блоків ДКП, які розбиваються на дві половини [3], і потім вбудовує секретний біт шляхом зміни різниці енергій між цими половинами. Енергія змінюється шляхом модифікації високочастотних коефіцієнтів ДКП. Алгоритм показує хороші результати в плані візуальної ідентичності оригінального і модифікованого сигналів, але складний в реалізації і вимагає багато ресурсів, що відбивається на часі його роботи. Таким чином, цей метод теж не був обраним в якості методу для реалізації.

Деякі методи складні в реалізації і виконуються досить довго, тому що забезпечують високу скритність вбудовування, деякі не підходять за іншими раніше визначеними певними вимогами. Наприклад, необхідна наявність первинних зображень або обсяг вбудованих даних занадто малий.

2.3.3 Метод вбудовування в область ДПФ

Дискретне перетворення Фур'є (ДПФ) - одне з фундаментальних понять в області обробки сигналів взагалі і цифрової обробки зображень зокрема.

Пряме перетворення Фур'є (Фур'є-образ) $F(u)$ неперервної функції однієї змінної $f(x)$ визначається рівністю:

$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-i2\pi ux} dx, \quad (2.1)$$

де i - уявна одиниця.

По заданому Фур'є-перетворенню $F(u)$ можна отримати вихідну функцію $f(x)$ за допомогою зворотного перетворення Фур'є:

$$f(x) = \int_{-\infty}^{+\infty} F(u)e^{-i2\pi ux}du. \quad (2.2)$$

Але в даному випадку нас цікавлять дискретні функції. Фур'є перетворення дискретної функції однієї змінної $f(x), x = 0, 1, 2, \ldots, M-1$, задається рівністю:

$$F(u) = \frac{1}{M}\int_{x=0}^{M-1} f(x)e^{-\frac{i2\pi ux}{M}}dx\,, u = 0, 1, 2, \ldots, M-1 \quad (2.3)$$

Це пряме дискретне перетворення Фур'є (ДПФ). Аналогічно, можна відновити вихідну функцію за допомогою зворотного ДПФ.

Для зображень, як для двовимірних масивів даних, актуальне використання двомірного ДПФ, тобто виконання ДПФ по рядках, а потім виконання ДПФ по результуючим стовпцях. При цьому бажано попередньо виконати центрування образу. В результаті, практично вся енергія концентрується в області низьких частот (візуально - центр), що корисно при високочастотній фільтрації зображень.

(а) (б)

Рисунок 2.2 - Зображення (а) і його Фур'є-образ (б). Спектр Фур'є представлений після застосування логарифмічного перетворення.

Запис ЦВЗ методом ДПФ передбачає роботу в частотному спектрі зображення і модифікацію значень таким чином, щоб при зворотному перетворенні вони не

призвели до помітних спотворень. Однак модифіковані значення повинні бути прочитані при повторному прямому Фур'є -перетворенні стего-зображення.

Складність такого методу полягає в тому, що в Фур'є-образ можливо записати тільки напів-крихкий або робастний ЦВЗ. При спробі запису крихкого ЦВЗ зворотне перетворення з наступним округленням значень може призвести до невеликих, але достатніх змін ЦВЗ, що спричинить руйнування неробастної мітки. Це варто враховувати при проектуванні способу формування і запису ЦВЗ.

Перевагою методу є можливість використання швидкого дискретного перетворення Фур'є (ШДПФ). Це дуже швидкий алгоритм обчислення ДПФ, що виконується за O (N log (N)). Це можливо завдяки «проріджуванню» - поділу процесу ДПФ на більш дрібні операції. Обмеженням при цьому є розмір ряду, по якому потрібно конвертувати. Розмір повинен бути кратним ступеню двійки 2k, де k - ціле число. Однак, в разі невиконання цієї умови можна доповнити ряд нулями, а потім врахувати це в результаті.

Робота в спектральної області зображення дозволяє гнучко управляти впливом мітки на результуюче зображення і записувати робастні ЦВЗ в області високих частот, тому що в цій області концентрується мінімум енергії (<10%). Це, в свою чергу, означає те, що для людини візуально оригінал і результат не відрізнятимуться - основна інформація про зображення концентрується в області низьких частот (максимум енергії). Для подальшої роботи було обрано саме цей метод.

2.4 Висновки до розділу 2

Сучасні стеганографічні методи розвиваються у двох основних напрямках:

1) методи, засновані на використанні спеціальних властивостей комп'ютерних форматів;

2) методи, засновані на надлишковості аудіо і візуальної інформації.

Методи цифрової та комп'ютерної стеганографії можна класифікувати за: цілями використання, видом обраного контейнера, структурою контейнера, за наявністю ключа, за способом обрання контейнера, за способом вилучення

вбудованого повідомлення, за способом організації контейнера та за способом доступу до інформації.

Серед методів вбудовування ЦВЗ можна відзначити просторові і частотні. Перші виконують запис знака безпосередньо в зображення, змінюючи значення кольоровості та інтенсивності окремих пікселів або блоків пікселів. Частотні методи працюють в спектральній області зображення. Були розглянуті найвідоміші методи вбудовування: метод заміни найменш значущого біту - він відмінно підходить у випадках, коли стего не буде зазнавати змін, тому що він крихкий, але за допомогою цього методу можна записати дуже велику кількість інформації. Дискретне косинусне перетворення показує хороші результати в плані візуальної ідентичності оригінального і модифікованого сигналів, але складний в реалізації і вимагає багато ресурсів, що відбивається на його швидкодії. Дискретне перетворення Фур'є дозволяє управляти впливом мітки на результуюче зображення і записувати робастні ЦВЗ в області високих частот, при цьому для людини візуально оригінал і результат не відрізнятимуться.

РОЗДІЛ 3. ПРОПОЗИЦІЇ ЩОДО МЕХАНІЗМУ ВНЕСЕННЯ ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ В ВІДЕО-КОНТЕНТ

Явище несанкціонованого розповсюдження контенту, захищеного авторськими правами, називається комп'ютерним піратством. Комп'ютерне піратство - незаконне розповсюдження або використання матеріалів, захищених авторським правом, без дозволу автора або з порушенням договору про використання таких матеріалів.

Такі дії є правопорушенням і переслідуються за законом. Однак, небажання деяких користувачів платити за працю автора призводить до того, що захищені авторським правом матеріали курсів стають публічно доступними необмеженому колу осіб.

Дана робота не наводить способи захисту інтернет-ресурсу від таких дій, але розглядає спосіб як скоротити витрати через швидку локалізацію джерела витоку, а саме акаунта користувача, який став поширювати медіа-дані купленого курсу.

Щоб розуміти, яким чином медіа-дані можуть бути скопійовані і поширені, слід трохи заглибитися в специфіку клієнт-серверної взаємодії - тобто процесу взаємодії пристрою користувача (студента) і сервера (онлайн-ресурсу), де студенту надано доступ до купленого освітнього курсу.

3.1 Клієнт-серверна взаємодія

Клієнт - це апаратний або програмний компонент обчислювальної системи, який посилає запити до сервера [34]. Клієнт може запитувати будь-які дані з сервера, маніпулювати ними безпосередньо на сервері, запускати на ньому нові процеси. Отримані дані клієнт може відображати користувачеві або використовувати іншим чином, в залежності від його призначення.

Сервер - програмний компонент обчислювальної системи, що виконує сервісні (обслуговуючі) функції по запиту клієнта [35]. Надає доступ до певних ресурсів або виконує обчислення.

У нашому випадку сервером можна вважати:

- саме фізичний пристрій, на яке в кінцевому рахунку через інтернет приходить запит від клієнта;

- веб-сервер, як програмне забезпечення, весь час запущене і очікує звернення від клієнта, щоб відразу ж обробити запит.

Клієнтом в такому випадку виступає:

- сам користувач, як клієнт компанії;

- комп'ютер або мобільний пристрій користувача, як фізичне пристрій;

- браузер користувача, як програма, безпосередньо формує запит на сервер і обробна його відповідь.

Браузер - прикладне програмне забезпечення для перегляду веб-сторінок, змісту веб-документів, комп'ютерних файлів та їх каталогів [36]; управління веб-додатками; а також для вирішення інших завдань.

Користувач може не тільки переглянути контент курсу в своєму браузері, а й зберегти його собі, щоб потім поширювати його на інших, непідконтрольних компанії майданчиках. Будемо вважати, що користувач уже сплатив курс і отримав необхідні дані для доступу до нього. Ці дані він вказує в своєму браузері, відбувається так званий процес «аутентифікації» - перевірка достовірності наданих користувачем даних для його авторизації.

Авторизація - це отримання прав доступу до ресурсів [37]. Таким чином, тепер користувач може в своєму браузері відкривати сторінки курсу і належних до нього уроків, а також переглядати весь необхідний контент.

При отриманні безпосередньо контенту взаємодія складається в такий спосіб:

1) користувач відкриває в браузері сторінку конкретного уроку будь-якого освітнього курсу;

2) браузер отримує html-сторінку уроку в якості відповіді на запит;

3) сторінка містить в собі сам текст уроку і безліч посилань на контент: зображення, відео, аудіо;

4) по знайдених посиланнях браузер автоматично знову посилає запит на сервер для отримання самих файлів;

5) проводиться завантаження файлів і їх відображення на сторінці, в залежності від типу файлу; наприклад, зображення відображаються «як є», а файли вбудовуються в «плеєр», який дозволяє керувати процесом відтворення відео.

Рисунок 3.1 Схема клієнт-серверної взаємодії при запиті сторінки, що містить медіа-дані

Якщо користувач трохи технічно підкований в цьому питанні, то розуміє, що ці файли, які відображаються на сторінці браузера, він може зберегти окремо і потім поширювати їх. Особливо гостро ця проблема проявляється для курсів, побудованих в форматі відеоуроків.

Повернемося до ідеї про те, як в такому випадку автор може себе захистити і мінімізувати втрати? Йому потрібно якомога швидше заблокувати обліковий запис порушника з метою обмежити його доступ до поточних і нових матеріалів, а також звернутися до правоохоронних органів, по можливості надавши дані про правопорушника, який був безпосередньо задіяний в незаконному розповсюдженні авторських матеріалів і повинен понести матеріальну відповідальність за свої дії.

Якщо кожна копія мультимедійних файлів будь-яким чином позначена і однозначно вказує на того користувача, кому вона була призначена, то не складає

труднощів, користуючись знанням про те, як цю мітку вилучити, знайти цього користувача в системі освітнього ресурсу і обмежити йому доступ до курсів, а також отримати про нього додаткову інформацію, яка може бути корисна правоохоронним органам.

Якщо згадати, яким чином контент потрапляє на комп'ютер користувача і відображається в браузері, ми приходимо до висновку, що мітка повинна бути безпосередньо в завантажуваному файлі. Це можливо, якщо використовувати принципи цифрової стеганографії.

3.2 Постановка мети

Таким чином, захист авторського контенту за допомогою застосування цифрової стеганографії є можливим. Досягається це проходженням певної послідовності дій:

1) генерація індивідуального для кожного користувача ЦВЗ («fingerprint», відбиток пальця);

2) вбудовування ЦВЗ в кожну копію мультимедійних файлів, запитуваних користувачем (наприклад, з веб-інтерфейсу платформи навчання);

3) вбудовування проводиться або в заздалегідь підготовлені копії, або в режимі реального часу - сервер транскодує мультимедійні файлі в режимі реального часу.

Принцип захисту спрацьовує в той момент, коли користувач робить нелегальне збереження авторського контенту і подальше його поширення. Вбудовані в клієнтські копії контенту водяні знаки дозволяють ідентифікувати його і прийняти супутні заходи.

Реалізацію ідеї захисту можна розділити на кілька великих блоків:

- платформа для навчання (вона ж - система доставки контенту; сайт, на якому безпосередньо знаходяться студенти під час вивчення курсів);

- підсистема захисту мультимедійних файлів, які надсилаються на клієнтську сторону (тобто студенту, хто переглядає в даний момент курс з браузера);

- підсистема визначення джерела «витоку» - виявлення мітки з медіафайлів і її декодування.

Сама по собі робота полягає в розробці алгоритму, який зможе виконати покладені на нього завдання приховання запису в даних і їх вилучення після проведення атаки, а також зробить можливим реалізацію бізнес-цінностей даного впровадження. Під бізнес-цінностями в даному випадку розуміється те, що алгоритм є частиною системи по боротьбі з незаконним поширенням авторського контенту, який продається клієнтам по моделі оплати доступу до мультимедійних файлів.

Повертаючись до змодельованої ситуації на прикладі освітніх ресурсів, відзначимо, що найбільший інтерес представляють такі медіа-дані, як зображення та відеофайли. Будь-яке відео являє собою набір кадрів. Тому відзначимо, що ілюстрації (зображення) - це частина відеоконтенту.

Відповідно, роботу алгоритму будемо розглядати і перевіряти саме на прикладі вбудовування ЦВЗ в зображення. Отримані знання і можливості потім можна екстраполювати на захист іншого контенту, в першу чергу, відеофайли.

Слідуючи поставленому завданню було визначено, що найкраще для захисту мультимедійних файлів шляхом впровадження ідентифікує записи підходять робастні ЦВЗ, тому що в процесі тиражування зображення або відео можуть бути ускладнені перешкодами або навмисне модифіковані. Робастні - значить «стійкі до спотворень» від англійського «robust» - міцний, твердий. Таким чином, метою даної роботи є розробка методу робастного захисту авторських мультимедійних файлів алгоритмами цифрової стеганографії на прикладі вбудовування ЦВЗ в зображення.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- виробити критерії для алгоритму вбудовування ЦВЗ (цифрового водяного знаку);

- порівняти методи вбудовування ЦВЗ і вибрати відповідний;

- розробити алгоритм захисту мультимедійних файлів на прикладі захисту зображень;

- представити прототип програмної реалізації алгоритму;

- протестувати розроблений алгоритм в умовах, наближених до реальних.

3.3 Критерії, що пред'являються до алгоритму

Для майбутнього алгоритму були сформульовані наступні вимоги:

1) висока швидкість роботи, на рівні реакції користувача (1 с);

2) візуальна ідентичність стего та оригіналу;

3) висока стійкість (робастність) для вилучення ЦВЗ із атакованих або стиснених медіафайлів;

4) можливість вилучення стего без наявності оригіналу.

Перша умова виходить з того, в якому середовищі передбачається використання алгоритму (веб-ресурси). Таким чином, робимо висновок, що ПЗ буде працювати в режимі реального часу, де дуже важливою є така метрика, як швидкість реакції користувача і швидкість відгуку. Як правило, перша реакція на запит користувача має надійти протягом секунди, інакше такий веб-додаток не можна назвати швидким і зручним для використання. Відповідно, протягом цієї секунди, коли сервер віддає запитану сторінку клієнту, необхідно підготувати контент і віддавати його на наступні запити від клієнта, що запросив сторінку (рис. 3.1)

Друга умова є класичною для стегосистем, адже головне її завдання - передати інформацію, приховавши при цьому сам факт передачі. Таким чином, виявити наявність мітки або її позицію неможливо, якщо оригінальне зображення і стего візуально не відрізняються. Плюсом цього також є те, що зображення залишається придатним для використання і прочитання, що теж дуже важливо, якщо повернутися до прикладу про освітні ресурси.

Висока стійкість необхідна для тих випадків, коли зловмисникові, який скомпрометував контент, відомо, що в зображеннях може бути мітка, або якщо він вирішив зменшити розмір зображень, збережених в форматі високої якості PNG, і перезберегти їх в форматі JPEG. У цьому випадку зображення дуже сильно ускладнюється шумами, що виникають в процесі роботи алгоритму стиснення JPEG, і мітка може виявитися пошкодженою. Робастний алгоритм (мітка) повинен перешкоджати цьому.

Заключна вимога є скоріше побажанням і рекомендацією, проте дуже сильно спрощує завдання пошуку “витоку”, якщо такий мав місце бути. Мається на увазі те, що якщо автор (компанія) виявив свій авторський контент в мережі, він може скористатися стегодекодером і відразу прочитати мітку. В іншому випадку довелося б шукати зображення-оригінал по базі ілюстрацій, які використовуються веб-ресурсом, який поширює цей контент, і потім порівнювати цей оригінал і знайдене зображення. Однак, і такий варіант не гарантує, що мітка буде коректно прочитана. Відповідно, вимога того, що мітка може бути прочитана однозначно без наявності контейнера-оригіналу є додатковим гарантом стійкості мітки (див. вимогу 3). Виходячи з описаних вимог потрібно визначити найбільш підходящий метод запису ЦВЗ за допомогою цифрової стеганографії.

3.4 Алгоритм вбудовування ЦВЗ

Був запропонований наступний алгоритм вбудовування ЦВЗ:

1) пряме перетворення Фур'є для зображення;

2) вбудовування сформованої мітки (ЦВЗ) в область високих частот спектральної області зображення;

3) зворотне перетворення Фур'є.

Такий нескладний алгоритм гарантує високу швидкість роботи ПЗ, яке забезпечує запис ЦВЗ в режимі реального часу.

Додатково для реалізації цього алгоритму потрібно сформувати мітку (ЦВЗ), придатну для використання в спектральній області і стійку до перешкод і різного роду атак, в тому числі стиску в форматі JPEG. Додатково варто відзначити обмеження, що накладаються цим методом. Перш за все, в нашому розпорядженні є не більше 80% контейнера для вкладення приховуваного повідомлення, тому що інші 20% містять максимум енергії, достатньої для відтворення зображення з мінімумом візуальних перешкод. Образ Фур'є представляє собою масив комплексних чисел, що містить в собі інформацію про амплітудний та фазовий спектри. Як показує практика, фазовий спектр несе дуже мало інформації про

кінцевий вигляд зображення і може бути легко змінений при модифікації контейнера, відповідно, для запису ЦВЗ ми можемо використовувати тільки амплітудний, щоб гарантувати робастний запис. Таким чином, залишається не більше 30-40% вихідної двовимірної площі для запису знака. Відповідно, знак повинен бути компактним, але при цьому зберігати властивість стійкості до перешкод.

Накладаються також і обмеження, що стосуються розміру зображення, придатного для запису ЦВЗ в нього. Виходячи з розміру мітки мінімальний розмір зображення може змінюватися.

Для зручності тут і далі будемо розглядати в якості ЦВЗ 32-бітне число, яке представляє собою ID користувача в системі, якому призначалася ця копія медіа-контенту.

Існує прямий взаємозв'язок між розміром секретного повідомлення і розміром мітки (ЦВЗ), якій воно представлено придатним для запису в контейнер.

Визначимо мінімальний розмір контейнера через певні раніше умови та інформацію про розмір мітки:

$$(m \times n) * 30\% < N_{\text{ЦВЗ}}, \tag{3.1}$$

де m та n - розмірність зображення, а $N_{\text{ЦВЗ}}$ - розмір мітки.

3.5 Алгоритм формування мітки

3.5.1 Характеристики мітки

32 біт досить для запису числа, що представляє собою більше 4 млрд.

Якщо використовувати тільки 16 біт із запропонованих 32, то можна записати максимальне число 65535, що часто для освітніх онлайн ресурсів є достатнім як позначення ідентифікатора користувача. Решту 16 біт можна використовувати для завадостійкого кодування, що буде корисно в умовах вимоги стійкості до зовнішніх змін.

Оскільки зображення - це набір пікселів, кожен з яких кодується по 1 байту на канал, то в цьому випадку ми маємо справу з дискретними значеннями, причому їх набір досить обмежений. У разі роботи з Фур'є-образом, ми маємо справу з комплексними числами, що вимагає точність обчислень порівнянну з обчисленнями значень з плаваючою комою. Однак, при зворотному перетворенні обов'язково пропадають деякі незначні дані в процесі округлення чисел з плаваючою комою. Відповідно, через проблеми округлення складно гарантувати, що пряме перетворення Фур'є, що піддалося потім зворотному перетворенню в зображення і знову прямому буде абсолютно ідентично.

Таким чином, маємо на увазі, що вбудована в спектральну область мітка гарантовано буде модифікована навіть зворотним перетворенням в зображення, тому що це обумовлено таким окремим випадком перетворення. При цьому не проводилося ніяких змін формату зображення або навмисних атак на стего.

Далі слід говорити про ЦВЗ і контейнер для ЦВЗ в термінах сигналів, тому що зображення - це сигнал аналогової природи для ока людини і працюємо ми з цим сигналом рівно так само, як з радіосигналом, сигналом від ехолокатора або будь-яким іншим, де перетворення Фур'є так само є одним з основних інструментів для аналізу і фільтрації даних.

Отже, спектральна область зображення являє собою набір коефіцієнтів, що характеризують амплітудні і фазові характеристики сигналу. Для різнорідного зображення (що не є однорідним) образ являє собою сигнал, схожий, в тому числі статистично, на випадкові дані.

Таким чином для записуваного ЦВЗ таке середовище, в яке ми його поміщаємо, можна трактувати, як середовище з високочастотним шумом. Стійким до таких умов є періодичний сигнал з правильно підібраними параметрами амплітуди і частоти. Вибір амплітуди важливий для того, щоб сильно не пошкодити зображення і не проявити мітку при зворотному перетворенні.

Вибір частоти дозволяє зробити сигнал стійким до прямих/зворотних перетворень Фур'є. Як правило, сигнал високої частоти проявляє себе найкраще. Прикладом тому може бути безліч технологій безпроводової передачі даних,

починаючи від цифрового радіо, закінчуючи 3G, LTE, WiMAX мережами, де частота вимірюється мегагерцами.

Однак, в наших умовах ми дуже обмежені розміром контейнера, а тому для дуже високочастотного сигналу його просто не вистачить, тому що тут ми стикаємося з ще одним фундаментальним поняттям в області обробки сигналів - теоремою Котельникова: будь-який аналоговий сигнал можна відновити з дискретного, якщо частота дискретизації удвічі більше максимальної частоти сигналу:

$$f > 2f_c \tag{3.2}$$

Коли основні параметри сигналів визначені, можна приступити безпосередньо до генерації мітки.

3.5.2 Генерація стійкої мітки

Як вже було визначено раніше, мітка являє собою бінарні дані розміром 32 біта. Самі по собі біти можуть бути представлені в термінах сигналів у вигляді:

- нижнього рівня (0);
- верхнього рівня (1).

Таким чином, запис 32-бітної мітки зводиться до кодування високих і низьких рівнів сигналу в будь-якій формі, придатній для протистояння перешкодам середовища, де ЦВЗ буде розміщений.

Як було зазначено раніше, високочастотний періодичний сигнал повинен бути стійким до таких перешкод, а це необхідно в разі читання Фур'є-образу зображення, що піддалося атакам. Таким чином, на даному етапі завдання зводиться до вираження 32-бітного слова (мітки) у вигляді періодичного сигналу. Це можливо реалізувати за допомогою техніки модулювання сигналу [38].

Модуляція - процес зміни одного або кількох параметрів високочастотного модульованого коливання за законом низькочастотного інформаційного сигналу [38]. Якщо спростити, то можна сказати, що вся управляюча інформація знаходиться в повідомленні - низькочастотному сигналі (модулюючий), а високочастотний

сигнал є модульованим. Коротка структурна схема проілюстрована на рисунку 3.2 Внаслідок модуляції бінарний дискретний сигнал записується в періодичний високочастотний за допомогою модуляції.

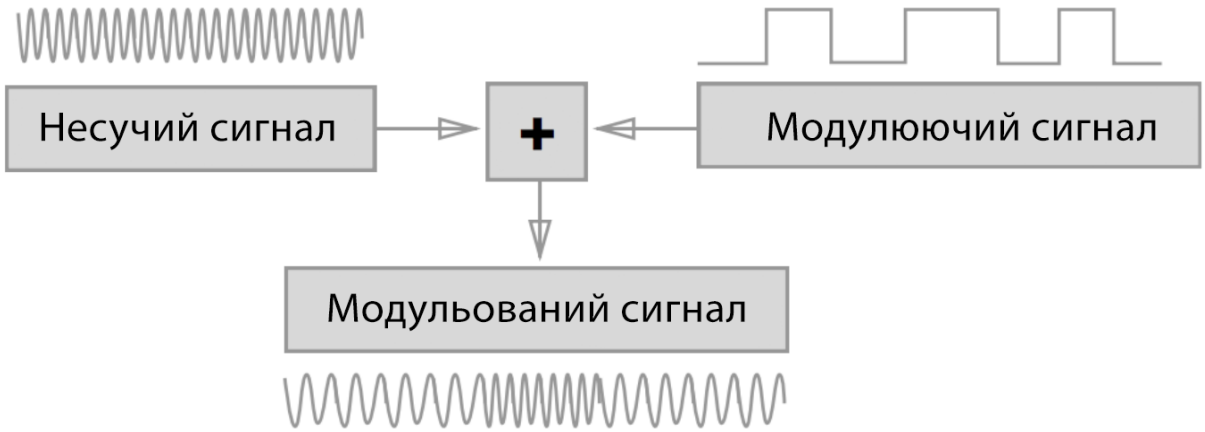

Рисунок 3.2 Ілюстрація принципу модуляції сигналу

Потрібно вибрати відповідний принцип модуляції.

Існує дуже багато методів модуляції і кожен по-своєму підходящий для конкретних випадків використання. Наприклад, раніше для передачі радіосигналу повсюдно використовувалася амплітудна модуляція сигналу (АМ). Передавати сигнал «як є» на далекі відстані було марно - природні джерела шуму, такі як атмосфера, дерева, рельєф, хмари і багато інших, дуже сильно псували сигнал. Було потрібно захищатися від перешкод. Принцип амплітудної модуляції полягає в тому, що керуючий сигнал управляє амплітудою несучого сигналу. Стикаємося з очевидними обмеженнями - амплітуда не може нескінченно зростати, а сигнал, однак, може містити велику кількість інформації, і якщо передавати її швидко, потрібен великий запас по амплітуді. Крім цього, такий спосіб хоч і протистояв різним природним перешкодам, все одно погано справлявся зі штучними, а також з подоланням стін будинків.

На заміну прийшла частотна модуляція сигналу (FM). У ній керуючий сигнал управляє частотою несучого, відповідно, чим вище рівень модулюючого сигналу, тим вище частота модульованого. Цей метод досить добре протидіє великій кількості шумів різного роду. Користуючись тим, що повсюдне практичне

використання такої технології виправдовує себе, при цьому в реалізації такий спосіб досить простий і швидкий в плані алгоритмічної складності, було вирішено вибрати для формування мітки саме цей принцип модуляції.

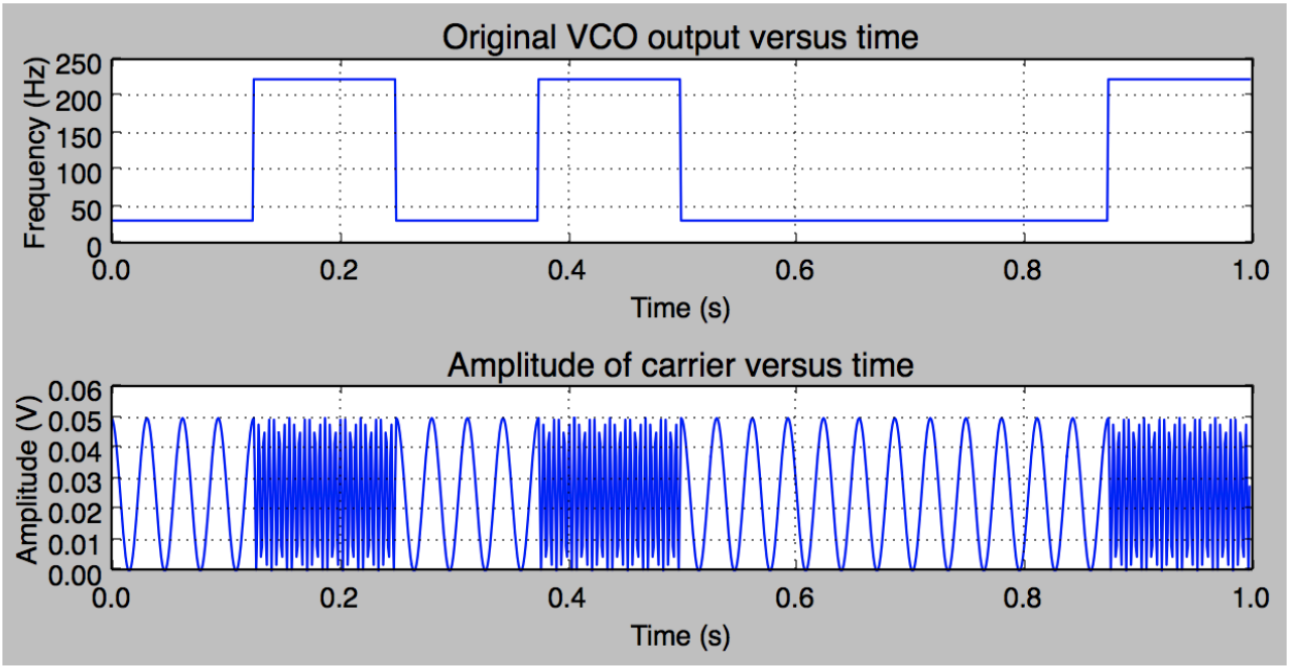

Рисунок 3.3 Частотна модуляція сигналу

Крім цього існує ще маса способів, специфічних для інших випадків використання. Серед них: різні види фазової модуляції, квадратурна модуляція, сигнально-кодова та інші.

3.5.3 Формування ЦВЗ, придатного для вбудовування

Згенерована мітка, що пройшла через модуляцію сигналу, може виявитися дуже довгою і непридатною для вбудовування в контейнер. Оскільки зображення являє собою двовимірний масив даних, то образ Фур'є виглядає ідентично. Відповідно, для вбудовування ЦВЗ ми оперуємо двовимірним контейнером, деякі області якого придатні для вбудовування. Як було зазначено раніше, необхідно записувати мітку в область високих частот спектра, тобто на кордонах двовимірної області, ближче до кутів.

Рисунок 3.4 Візуалізація модульованого бінарного слова

Тоді потрібно представити довгий числовий ряд, в якому зараз записана мітка, в більш компактний вигляд. Це можливо, якщо просто розбити довгий рядок на кілька коротших. Внаслідок цього мітка теж виглядає як двовимірний масив даних. Відновлення мітки проводиться в зворотному порядку - спочатку читаємо масив даних з спектральної області, а потім порядково розгортаємо його і працюємо, як з оцифрованим періодичним аналоговим сигналом, застосовуючи всі доступні інструменти аналізу.

Але згорнута в двовимірний масив мітка може знову бути піддана дії шумів, тому що буде представляти із себе неперіодичний високочастотний сигнал, причому при зворотному перетворенні породжуючий осциляції на результуючому зображенні.

Було знайдено рішення у вигляді вертикального порядкового дублювання. Тобто рядки в двовимірному масиві дублюються k разів, щоб потім при відновленні мати можливість взяти середнє значення m для кожного значення рядка, що шукається:

$$m_i = \sum_{i=1}^{k} n_i \tag{3.3}$$

Ілюстрація вихідного ряду, згорнутого в двовимірний масив і підданого дублюванню рядків, наведена на рисунку 3.5.

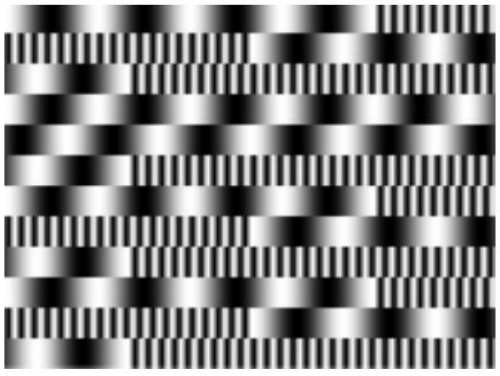

Рисунок 3.5 Візуалізація двовимірного масиву фазово модульованого бінарного сигналу з дублюванням рядків проти перешкод

3.6 Висновки до розділу 3

У третьому розділі були розглянуті основні визначення, що стосуються клієнт-серверного взаємодії, проводиться занурення в предметну область цифрової стеганографії й детальніше розкривається проблематика роботи. Після розгляду клієнт-серверної архітектури приходимо до висновку, що мітка повинна розміщуватися безпосередньо в файлі, який собі завантажує користувач.

Далі були сформульовані наступні вимоги до майбутнього алгоритму:

1) висока швидкість роботи, на рівні реакції користувача (1 с);

2) візуальна ідентичність оригіналу та модифікованих даних;

3) висока робастність для вилучення ЦВЗ зі змінених медіафайлів;

4) можливість вилучення стего без наявності оригіналу.

В даному розділі був запропонований алгоритм вбудовування ЦВЗ:

1) пряме перетворення Фур'є для зображення;

2) вбудовування сформованої мітки (ЦВЗ) в область високих частот спектральної області зображення;

3) зворотне перетворення Фур'є.

Був також описаний метод генерації мітки для вбудовування. Було обрано фазову модуляцію для формування мітки, в реалізації такий спосіб модуляції є досить простим і швидким в плані алгоритмічної складності. Для подальшого вбудовування ЦВЗ мітка повинна пройти через процедуру вертикального порядкового дублювання, коли рядки в двовимірному масиві дублюються k разів, щоб потім при відновленні мати можливість взяти середнє значення m для кожного значення рядка.

РОЗДІЛ 4. ПРАКТИЧНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ВБУДОВУВАННЯ ТА ВИЛУЧЕННЯ ЦИФРОВОГО ВОДЯНОГО ЗНАКУ З ВІДЕОКОНТЕНТУ

4.1 Вибір інструментарію

В якості мови для розробки був обраний Python - високорівнева мова з мінімалістичним синтаксисом, сфокусована на продуктивності розробника і читанні коду. Такий вибір був обумовлений великою кількістю наукових бібліотек, які дозволяють працювати з зображеннями, сигналами, частотної областю, графіками. Дуже поширений в науковому середовищі, як середовище, що не вимагає глибокого вивчення для того, щоб почати писати код. Існує велика кількість сторонніх бібліотек і пакетів, які розширюють стандартну функціональність мови. Це і багатопотокова робота, і робота з віддаленими серверами, і робота з великими масивами даних, а також астрономічні розрахунки, зручна робота з зображеннями та побудова графіків.

До речі, останнє є очевидною перевагою мови й екосистеми Python в цілому перед іншими скриптовими аналогами. Конкуренцію цій мові становить мова R, але вона має вищий поріг входу і націлена в більшій мірі саме на наукові розрахунки і обчислення, в той час як Python є мовою широкого профілю та застосовується в багатьох областях: веб-додатки, розрахунки, GUI-додатки, боти, консольні утиліти та багато іншого. Саме це дає ще один плюс на користь вибору даної мови, тому що розроблений прототип буде придатним для вбудовування в працюючі системи - таке впровадження коштує дуже дешево.

Для розробки прототипу було обрано такі бібліотеки:

- numpy - робота з різними типами чисел і масивами даних;
- scipy.signal - цифрова обробка сигналів;
- pillow - робота з зображеннями;
- matplotlib - зручне побудова графіків.

Даний набір бібліотек використовувався не тільки для прототипування і дослідження, а також для основної реалізації. Всі представлені в роботі графіки і дослідження були зроблені за допомогою даних інструментів. Що зручно, такі речі як ДПФ і швидке ДПФ вже реалізовані і включені в пакет sciру. Крім цього, в них є ще більша кількість різних інструментів для аналізу сигналів і роботи з різними даними.

NumPy - це бібліотека з відкритим вихідним кодом для мови програмування Python. Вона має підтримку багатовимірних масивів (включаючи матриці) і високорівневих математичних функцій, призначених для роботи з багатовимірними масивами.

Pillow - бібліотека мови Python, призначена для роботи з графікою. Вона дозволяє працювати з великою кількістю форматів зображень. Пакет pillow є форком (відгалуженням) відкритого проекту PIL (Python Imaging Library) і розширює його функціональність, а також адаптує роботу бібліотеки для нових версій Python. Серед можливостей пакету: читання зображень різних форматів (PNG, JPEG, TIF, GIF і ін.); збереження зображень в ці формати; розбір зображень на складові (кольорові зображення на канали); робота з зображенням, як з двовимірним масивом даних. Що зручно, дані зображення легко перетворюються в типи даних, з якими працює пакет numpy, і назад. Немає необхідності докладати додаткові зусилля для візуалізації двовимірного масиву даних.

Matplotlib - бібліотека на мові програмування Python для візуалізації даних двовимірної (2D) графікою (3D графіка також підтримується). Пакет matplotlib включає потужні засоби для візуалізації наборів даних у вигляді графіків і діаграм. Використані в даній роботі ілюстрації роботи алгоритму були отримані саме за допомогою можливостей цього пакету.

SciPy - бібліотека для мови програмування Python з відкритим вихідним кодом, призначена для виконання наукових та інженерних розрахунків. Дана бібліотека включає в себе широкий функціонал для обробки сигналів та зображень. Також в ній міститься бібліотека для статистичних розрахунків. Пакет sciру являє собою цілу екосистему, що включає всі перераховані вище пакети, а також

однойменну бібліотеку ядра, що містить, в тому числі, scipy.signal - набір інструментів для цифрової обробки сигналів. Це необхідно для роботи алгоритму модуляції / демодуляції.

Підключення бібліотек у скрипті виконується наступними інструкціями:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as signal
import scipy.signal.signaltools as sigtool
from PIL import Image
```

4.2 Реалізація розробленого алгоритму

4.2.1 Генерація ЦВЗ

Оскільки не пред'являється вимога високої скритності передачі (маскування мітки в частотної області), то мітка жодним чином не залежить від контейнера, в який вона вбудовується, а стегошлях є тривіальним - необхідно лише заздалегідь визначити відступ від кордонів двовимірного контейнера і запам'ятати його для подальшого читання мітки.

Відповідно, підсистеми для генерації ЦВЗ і для вбудовування згенерованого ЦВЗ працюють незалежно - унікальна мітка користувача може бути підготовлена заздалегідь, а може бути створена в процесі роботи основного алгоритму.

Саме тому в даній роботі ці частини описані в різних підрозділах.

Отже, раніше ми визначили, що об'єктом вбудовування (секретним повідомленням) буде 32-бітне слово, яке, наприклад, містить ідентифікатор користувача онлайн-ресурсу, якому призначається дана копія мультимедійних файлів.

Для зручності, будемо розглядати деяку випадкову мітку sgn:

```
sgn = 0b10101001011011101011001011111010 (32 bits)
```

Префікс «0b» означає бінарне представлення даних у багатьох мовах програмування, і в Python в тому числі.

Як було описано в теоретичній частині, бінарні дані повинні бути представлені у вигляді періодичного сигналу (наприклад, синусоїдальна хвиля).

Досягається це за рахунок методів модуляції сигналу, в нашому випадку - за рахунок частотної FM модуляції, яка часто використовується для безпроводової передачі даних. Даний метод модуляцій придатний і для аналогового сигналу, але в нашому випадку будемо моделювати тільки значення 0 і 1, що означають низький і високий рівень сигналу відповідно. Такий окремий випадок називається частотною маніпуляцією або Frequency Shift Keying (FSK) в англомовній літературі [35].

Процес запису цифрового сигналу в періодичний називається модуляцією, а зворотний процес - демодуляцією. Виконують ці операції модулятор і демодулятор відповідно, а пристрій, що об'єднує в собі обидві ці функції називається модемом (modem - modulation / demodulation).

У загальному вигляді роботи системи модуляції / демодуляції виглядає, як зображено на наступній схемі (рис. 4.1).

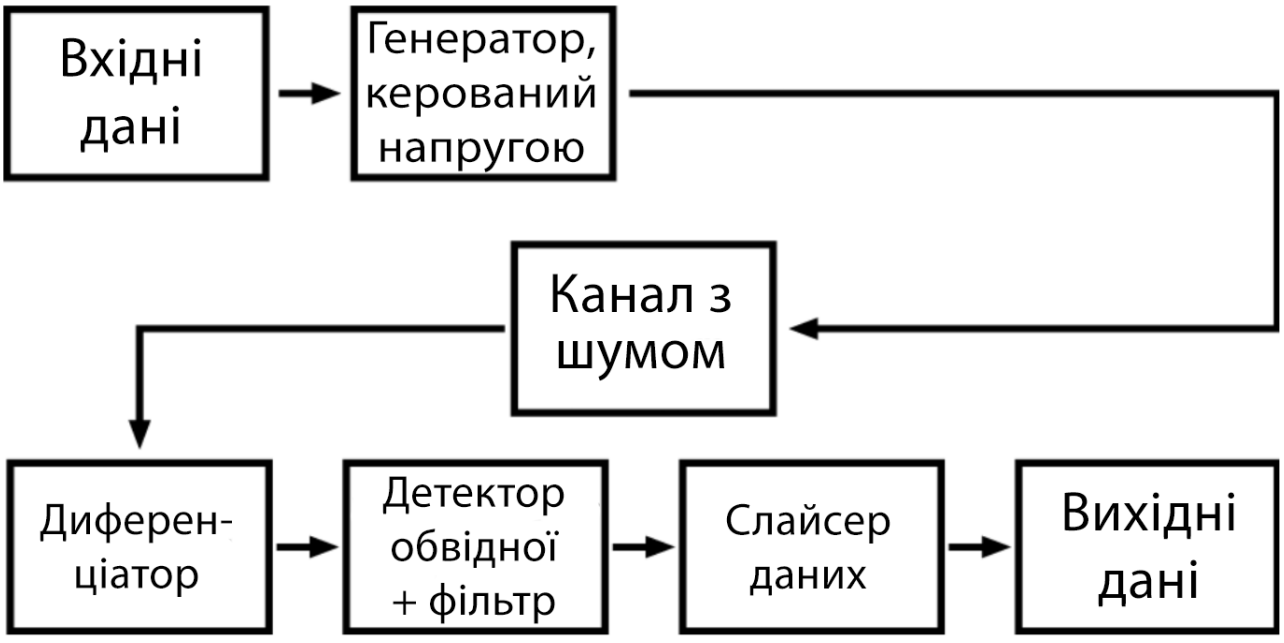

Рисунок 4.1 Схема роботи системи модуляції / демодуляції

Спочатку формуються і готуються модульовані дані (керуючий сигнал), а безпосередньо генерацію змодульованого сигналу виконує синтезатор або по-іншому генератор, керований напругою (VCO, Voltage-Controlled Oscillator). Це електронний генератор, частота коливань якого управляється поданою на вхід напругою. У нашому випадку є всього 2 рівня напруги, відповідно, пристрій буде генерувати 2 частоти. В результаті, отримуємо періодичний сигнал синусоїдальної форми, що складається з наборів відрізків двох частот.

Заздалегідь потрібно підібрати параметри генератора: частота для верхнього рівня, частота для нижнього, період (довжина) відрізка для кодування одного біта (швидкість). Виходячи з цих параметрів можна визначити довжину майбутнього сигналу.

Як було згадано в теоретичній частині, висока частота дозволяє сигналу зберігати свої характеристики в зашумленому середовищі. Чим вище частота сигналу - тим вище повинна бути частота дискретизації, а значить для кодування одного біта потрібно зберігати більше даних. Крім цього, в залежності від бітрейту, тобто кількості даних на одиницю часу, розмір сигналу для кодування одного біта також може змінюватися.

Це варто враховувати при розрахунку граничних значень використання алгоритму. Можна зробити висновок про те, що чим більший контейнер є в нашому розпорядженні, тим більше даних ми можемо записати, а відповідно, можна або посилити робастність стеганографічної мітки, або збільшити обсяг приховуваного повідомлення (наприклад, збільшити до 64 біт і так далі).

Отже, чим вищий бітрейт і частота сигналу - тим він стійкіший до перешкод, але займає більший обсяг даних при кодуванні.

Позначимо сигнал для кодування та його характеристики через формулу (4.1):

$$m(t) = \begin{cases} 0: & +f_{dev}, \\ 1: & -f_{dev}. \end{cases} \tag{4.1}$$

де $m(t)$ - кодований біт в момент часу t; f_{dev} - frequency deviation, девіація частоти відносно базового значення; $f_c - f\ carrier$, несуча частота в залежності від кодованого біта.

Наведемо цикл роботи синтезатора сигналу в умовах даного завдання:

1) визначення майбутньої довжини сигналу і побудова тимчасового ряду *t*;

2) формування повідомлення із відповідності значення частоти значенням рівня (нижній або верхній) згідно з формулою (4.1);

3) дублювання значень повідомлення відповідно до бітрейту, при цьому довжина ряду повинна бути еквівалентною часовому;

4) побудова періодичної хвилі через застосування функції синус / косинус над рядом, що представляє собою добуток часового ряду та повідомлення, представленого у вигляді набору значень частот (4.2).

В результаті отримуємо ряд *y*, що представляє собою дискретний запис частотно-модульованого сигналу періодичної хвилі, що несе в собі бінарне повідомлення (4.2).

$$y = Acos(2\pi \cdot (f_c + m(t)) \cdot t) \tag{4.2}$$

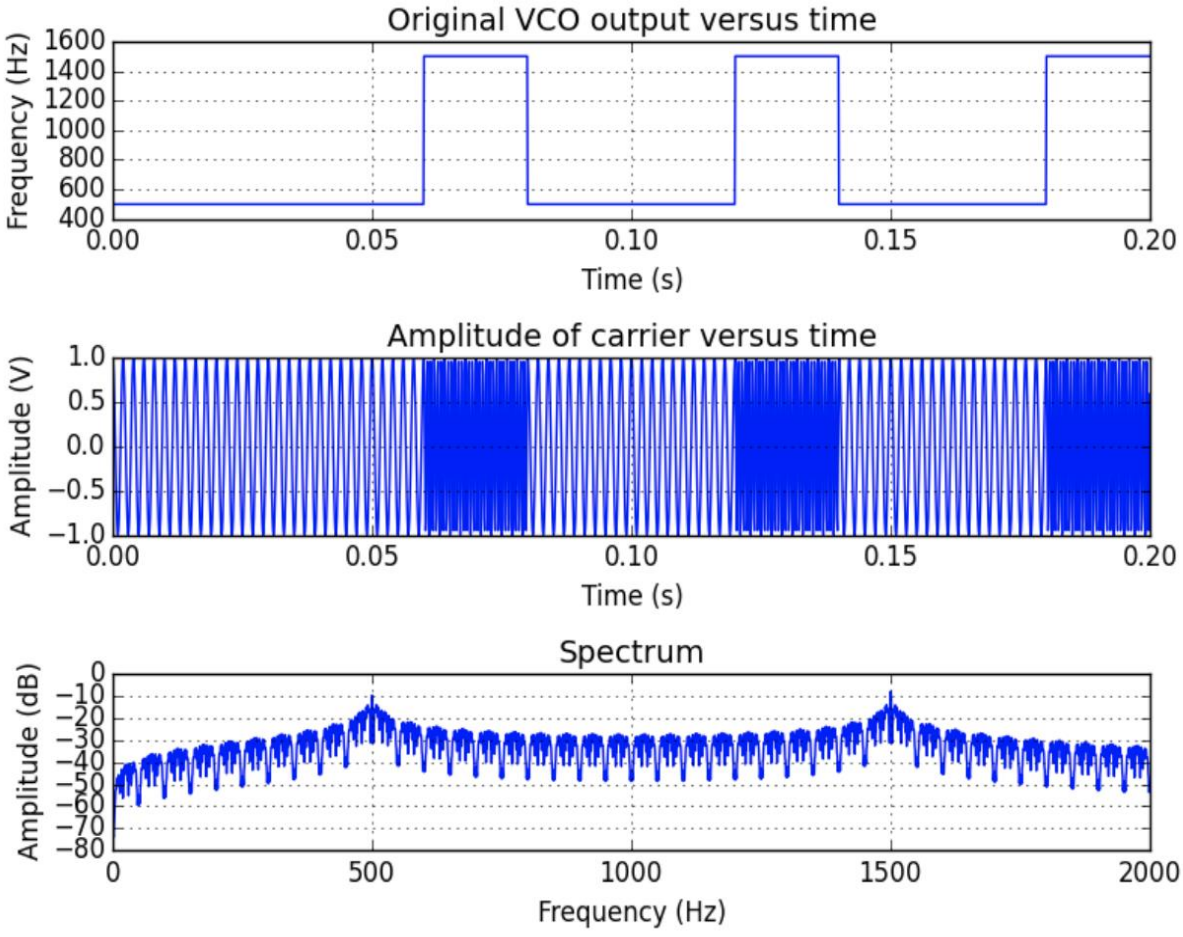

Рисунок 4.2 Ілюстрація роботи синтезатора фазової маніпуляції

Резюмуючи, для даного етапу потрібно визначити наступні характеристики кодування сигналу:

- несуча частота f_c;
- відхилення частоти f_{dev};
- бітрейт і частота дискретизації F_c;
- амплітуда A.

Кілька слів про останню характеристику: амплітуда жодним чином не впливає на довжину сигналу або алгоритмічну складність його запису, проте вона є важливим елементом безпосередньо при вбудовуванні ЦВЗ. Відповідно, на даному етапі амплітуду можна взяти за одиницю і отримувати в результаті нормалізований сигнал, який далі можна модифікувати за амплітудою в залежності від умов використання. Наприклад, висока амплітуда для ЦВЗ в частотному спектрі зображення може викликати видимі спотворення при зворотному перетворенні.

Результат роботи синтезатора відображений на рисунку 4.2.

Інструкції, що реалізують роботу синтезатора, на мові Python:

```
N = 32 # довжина вбудованого повідомлення
Fs = 10000 # частота дискретизації
Fc = 1000 # несуча частота
Fdev = 500 # відхилення частоти
bitrate = 50 # бітрейт
A = 1 # одинична амплітуда

# генерація випадкової мітки
data_in = np.random.random_integers(0,1,N)
# генерація часового ряду
t = np.arange(0,float(N)/float(bitrate),1/float(Fs),
dtype=np.float)
# генерація ряду значень частоти від часу
m = np.zeros(0).astype(float)
for bit in data_in:
```

```
        if bit == 0:
             m=np.hstack((m,np.multiply(np.ones(Fs/bitrate),Fc-
        Fdev)))
        else:
             m=np.hstack((m,np.multiply(np.ones(Fs/bitrate),Fc+Fdev)
        ))
    # генерація періодичного сигналу
    y=np.zeros(0) y=A * np.cos(2*np.pi*np.multiply(m,t))
```

4.2.2 Вбудовування ЦВЗ

Безпосередньо вбудовування згенерованої мітки виконується тривіальним способом, що дозволяє алгоритму залишатися дуже швидким і придатним для використання в масштабі реального часу. Як було зазначено раніше, головною характеристикою на даному етапі є амплітуда сигналу, яку можна легко регулювати й експериментально підбирати для мінімізації візуальних перешкод при зворотному перетворенні. Але спектральна область зображення з наступними прямими-зворотними перетвореннями представляє для ЦВЗ середовище з великим рівнем шуму. Відповідно, амплітуда повинна бути підібрана таким чином, щоб ЦВЗ читався після цих перетворень, але не впливав на саме зображення.

Крім цього, необхідно заздалегідь визначити позицію для запису мітки, однакову для всіх зображень, що піддаються стеганографічному перетворенню. Наприклад, можна взяти відступ в 10 пікселів для кожної з двох сторін правого верхнього кута двовимірного масиву Фур'є-образу зображення. Це залишить запас для боротьби з таким видом атаки на зображення, як обрізка країв. Накладення шуму на ЦВЗ можна представити у вигляді формули (4.3):

$$y = Acos\big(2\pi \cdot \big(f_c + m(t)\big) \cdot t\big) + n(t) \quad (4.3)$$

де $n(t)$ - значення шуму, залежне від часу (від позиції конкретного значення ряду вбудованого ЦВЗ).

Цикл вбудовування описується наступними діями:

1) ряд *y* вихідного сигналу згортається в двомірний масив за допомогою розбиття ряду на відрізки заздалегідь визначеної довжини з урахуванням необхідної компактності мітки;

2) мітка вставляється в заздалегідь визначену позицію спектральної області шляхом заміни значень;

3) зворотне перетворення зображення з частотного спектру у видимий.

При цьому вже при зворотному перетворенні відбувається накладення шуму на мітку. При різного роду атаках на зображення, в тому числі стиснення зображення в іншому форматі, шум посилюється і пошкоджує мітку сильніше.

На рисунку нижче проілюстровано процес запису мітки в Фур'є-образ зображення.

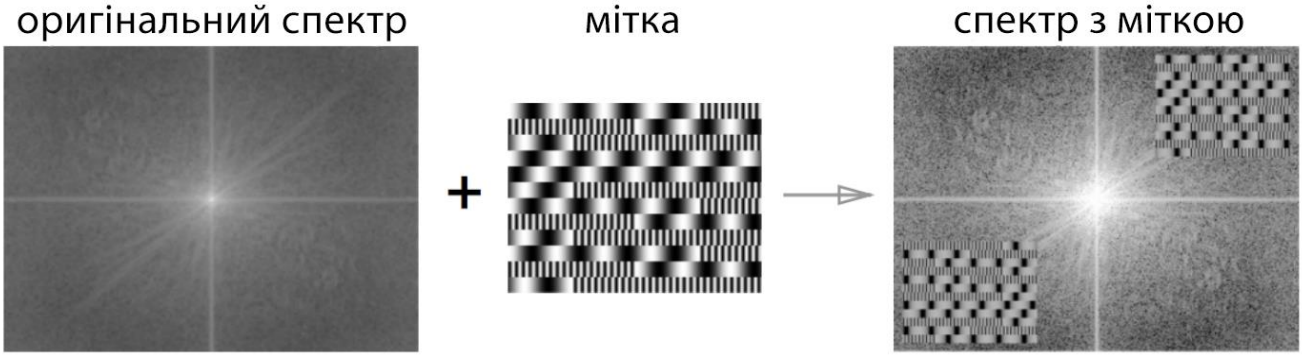

Рисунок 4.3 Ілюстрація процесу запису мітки в Фур'є-образ

Результати читання зображені на рисунку 4.4.

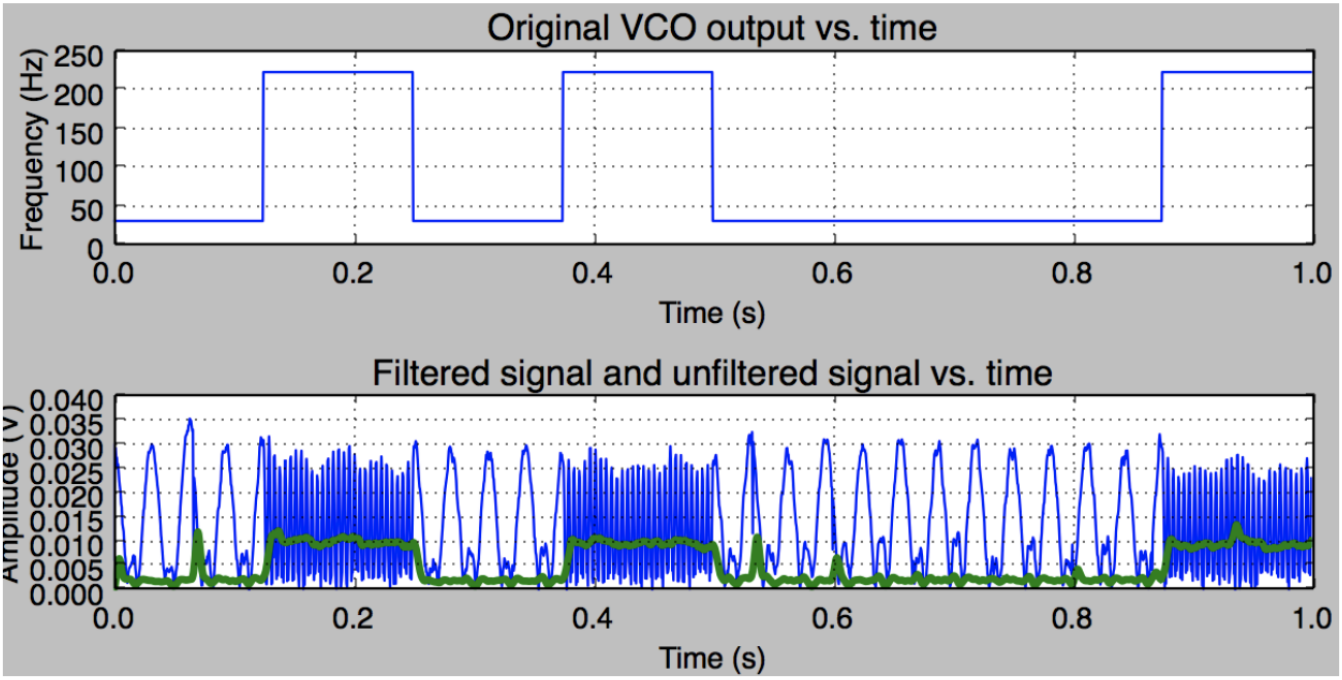

Рисунок 4.4 Сигнал мітки з накладеним шумом після прямого перетворення в зображення формату PNG

Інструкції на мові Python для подання мітки у вигляді двовимірного масиву і її запису в частотну область:

```
# читання оригінального зображення
imgSource = Image.open('test/house.png').convert("RGB")
imgArr = np.asarray(imgSource)
(hei, wid, chls) = np.shape(imgArr)

# генерація шару з міткою для накладення
overlay = sgn padTop,
padLeft = tuple(np.subtract(np.shape(imgArr[..., 0]),
np.shape(overlay)))
overlay = np.pad(overlay,
      ((y_offset, padTop-y_offset),
      (padLeft-x_offset, x_offset)),
      'constant', constant_values=0.0)
```

```
# мітка дублюється симетрично відносно центру
overlay = overlay + np.fliplr(np.flipud(overlay))

# запис шару з міткою в кожен з каналів
channels = (
     np.where(overlay != 0, overlay, fft(imgArr[..., 0])),
     np.where(overlay != 0, overlay, fft(imgArr[..., 1])),
     np.where(overlay != 0, overlay, fft(imgArr[..., 2])),
)

# зворотне перетворення
channels = (
     np.clip(ifft(channels[0]), 0, 255),
     np.clip(ifft(channels[1]), 0, 255),
     np.clip(ifft(channels[2]), 0, 255),
)

# збереження зображення в файл
Image.fromarray(np.uint8(np.dstack(channels)))
.save('results/out.png')
```

4.2.3 Читання ЦВЗ

З рисунку 4.1 можна зробити висновок, що читання мітки є складнішим процесом, ніж її генерація і вбудовування. Пов'язано це з тим, що сигнал потрібно демодулювати, буквально «вгадати» його значення в певний момент часу. Коли сигнал незначно ускладнений шумами, його легко прочитати як візуально, так і програмно. У разі роботи в середовищі з високою амплітудою і частотою шуму необхідно озброїтися інструментами статистичного аналізу для правильного визначення значення функції в даний момент часу.

Згідно з вищенаведеною схемою, читання сигналу можна розбити на наступні модулі:

1) диференціатор;

2) детектор конвертів + фільтр;

3) нарізка;

4) висновок результуючих даних.

На кожному етапі використовуються проміжні значення, отримані при виконанні попереднього етапу. Перш за все, вилучення мітки починається безпосередньо з читання зображення. Потім проводиться читання двовимірного масиву мітки із заздалегідь визначеного положення. Зауважимо, що розмір мітки також повинен бути заздалегідь відомий. Він визначається раніше озвученими характеристиками, такими як частота дискретизації, бітрейт та іншими.

Читання мітки включає в себе такі кроки передобробки, як:

1) читання двовимірного масиву;

2) усереднення значень по кожному рядку, використовуючи коефіцієнт дублювання *k*;

3) розгортання масиву в один рядок - ряд даних сигналу.

На отриманому ряді прочитаного сигналу виконує свою роботу диференціатор - модуль для обчислення похідної першого порядку. Це необхідно для відділення сигналу даних від несучої частоти. Принцип роботи диференціатора описаний у формулі (4.4):

$$\frac{dy}{dt} = -A2\pi\left(f + m(t) + t\frac{dm(t)}{dt}\right)\sin\left(2\pi * \left(f_c + m(t)\right) * t\right) + \frac{dn(t)}{dt}, \qquad (4.4)$$

де $n(t)$ - функція шуму.

Це обчислення можна спростити і прискорити користуючись тим знанням, що можна ігнорувати фазовий зсув, тому що він є постійним. Однак, якщо ми не знаємо, де знаходиться мітка або вона була зміщена, необхідно застосовувати біти синхронізації для відновлення фазового зсуву. Для спрощення візьмемо його за константу. В результаті такого припущення, справедливого для даного випадку використання, $dm(t)/dt$ дорівнює нулю, і цю частину можна опустити.

Спрощений вираз має вигляд:

$$\frac{dy}{dt} = A\big(f + m(t)\big)\sin\big(2\pi * \big(f_c + m(t)\big) * t\big) + \frac{dn(t)}{dt} \quad (4.5)$$

Далі працює детектор конвертів (детектор огинаючої) - пристрій для відділення високочастотного несучого сигналу від низькочастотного сигналу повідомлення. Фільтр, що використовується - низькочастотний, він залишає тільки необхідні нам значення низької частоти, що несе в собі корисну інформацію.

Результат фільтрації сигналу запишемо в *y_filtered*:

$$y\_filtered = A\big(f + m(t)\big) + \frac{dn(t)}{dt} \quad (4.6)$$

Заключним етапом є «нарізка» даних відповідно до заздалегідь визначеного розміру даних для кодування одного біта. Крім цього на цьому етапі виконується компенсація постійного амплітудного зміщення (DC), що дозволяє деяким чином компенсувати накладення шуму на сигнал. Для всього ряду визначається значення медіани (mean), яке використовується для прийняття рішення під час декодування бітів. Цикл роботи модуля прийняття рішень:

1) нарізка ряду на періоди, що відповідає одному біту даних;

2) вибір значення в середині періоду і порівняння з медіаною.

Якщо значення вище медіани, то воно трактується як одиниця в іншому випадку - нуль. Таким чином, в результаті ми отримуємо демодульоване повідомлення.

Значення на етапі нарізки можна визначити наступною формулою:

$$y\_slicer = m(t) + \frac{dn(t)}{dt} \quad (4.7)$$

Оригінальний бінарний сигнал, його частотна модуляція в середовищі з низьким рівнем шуму і результат роботи диференціатора й подальшої фільтрації наведено на рисунку 5.4. Лінія на графіку частотно модульованого сигналу являє собою відфільтрований сигнал, по ньому і будується медіана.

Те, як впливає високоамплітудний і високочастотний шум на оригінальний сигнал, можна поспостерігати на рисунку 4.5.

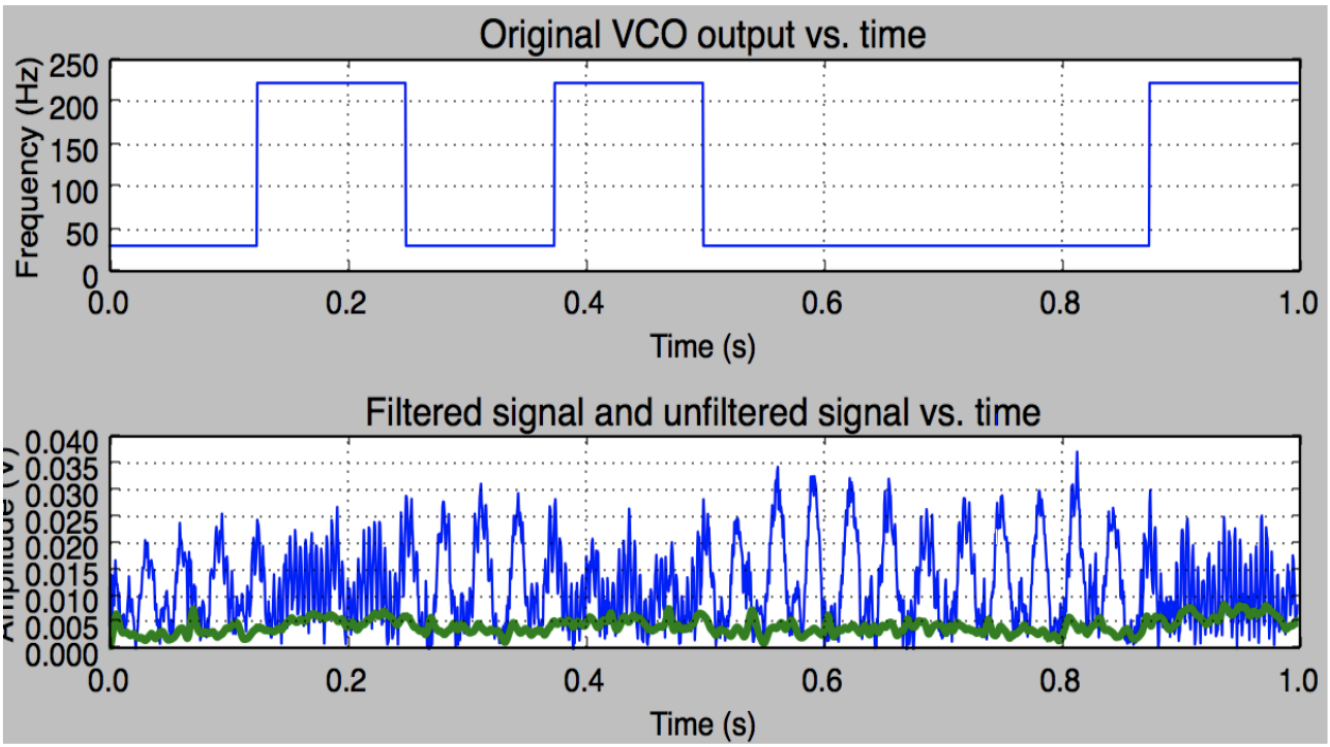

Рисунок 4.5 Сигнал мітки з накладеним шумом після стиснення оригінального зображення з формату PNG в формат JPG

Інструкції, що реалізують роботу вищеописаних модулів:

```
# читання зображення зі стего
test = np.asarray(Image.open('results/input.jpg').convert("L"))

# пряме перетворення Фур'є
saved = fft(test)

# «вирізання» мітки із завчасно відомої позиції
restored = np.zeros(stego_row).astype(float)
received = saved[slicing]
for i in range(0, height/repeats):
     restored = np.vstack((restored,

# порядкове усереднення значень
np.average(received[i*repeats:i*repeats+repeats], axis=0)))
restored = restored[1:]
restored = np.abs(np.reshape(restored, (signal_len, )))
```

```
 restored = restored - np.amin(restored)

 # диференціатор
 y_diff = np.append(np.diff(restored,1), [.0])

 # low-pass фільтрація і визначення обвідної
 y_env = np.abs(sigtool.hilbert(y_diff))
 h=signal.firwin(numtaps=numtaps,    cutoff=bitrate*2,    nyq=Fs/2)
 y_filtered=signal.lfilter(h, 1.0, y_env)

 # обчислення медіани
 mean = np.mean(y_filtered)

# прийняття рішення по серединам періодів біт
rx_data = []
sampled_signal = y_filtered[Fs/bitrate/2:len(y_filtered):Fs/
bitrate]
for bit in sampled_signal:
     if bit < mean:
        rx_data.append(0)
     else:
        rx_data.append(1)

# підрахунок кількості помилок
bit_error=0
for i in range(0,len(data_in)):
     if rx_data[i] != data_in[i]:
          bit_error+=1
error_percentage = (float(bit_error)/float(N)*100)
```

Для перевірки результатів роботи алгоритму була використана оцінка кількості помилок в процентному повідомленні. Компенсацію помилок на цьому етапі можна забезпечити за рахунок завадостійкого кодування. Наприклад, використовуючи коди Ріда-Соломона. У даній роботі ми не будемо зупинятися на

цьому докладно, тому що це тема для окремої розмови, і ми просто можемо вільно використовувати наявні напрацювання в потрібній нам комбінації.

4.3 Тестування розробленого алгоритму

Для тестування роботи алгоритму і його програмної реалізації у вигляді прототипу на Python було вирішено використовувати середовище, близьке до реальних випадків використання такого рішення.

В якості «атакуючого» середовища була обрана соціальна мережа «Facebook» і популярний месенджер «Телеграм». Такий вибір був зроблений тому, що ці сервіси є дуже популярними для передачі даних, публікації контенту, його зображення і поширення. Відповідно, вони мають повний набір зручних інструментів для завантаження й управління медіа-даними. У їх числі і зображення, і відео, і анімовані зображення (.gif).

У кожній з цих мереж в якості формату для зберігання зображень за замовчуванням використовується JPEG, при цьому у Facebook зображення стискаються сильніше і розмір оригінального PNG може бути стиснений в 10 разів.

Завдання прототипу - не зважаючи на втрату великої кількості надлишкової інформації після стиснення, відновити й прочитати мітку.

4.3.1 Практичні тести в реальних умовах

Для тестування були обрані стандартні в області тестування алгоритмів для роботи з зображеннями ілюстрації: cameraman, house, lena, peppers. Частина з них - монохромні, частина - кольорові. Перший тест - на зображенні cameraman (рисунок 4.6).

Рисунок 4.6 Етапи перетворення зображення cameraman

Як видно з рисунку, оригінал зображення несуттєво змінився в розмірі після вбудовування мітки. Після завантаження зображення в соцмережу, воно було перезбережене у форматі JPEG і зменшилося в розмірі в 5 разів.

Однак, декодером мітка була прочитана без помилок. Аналогічний результат був продемонстрований і для зображення, завантаженого в месенджер «Телеграм».

Другий тест - на зображенні house (рисунок 4.7).

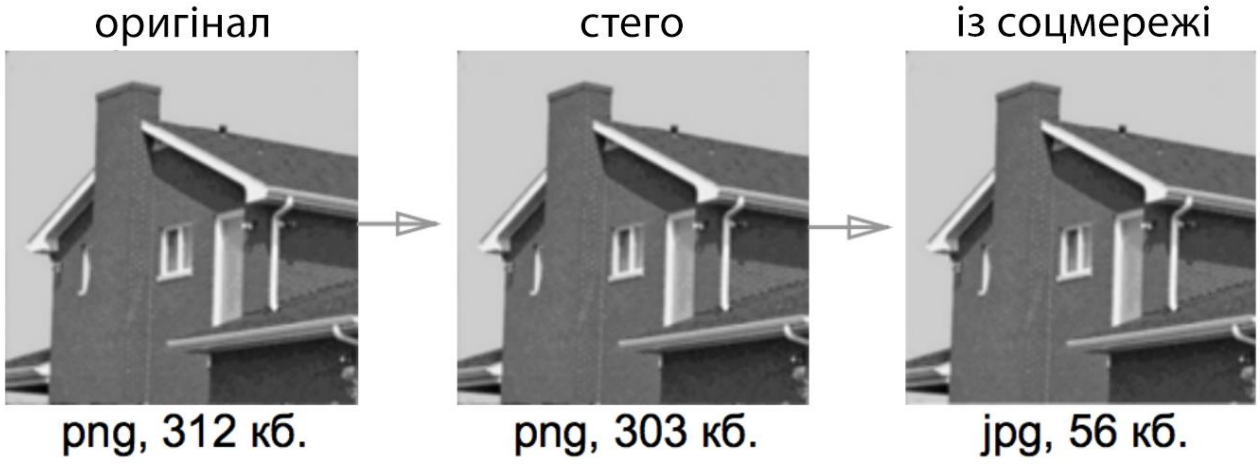

Рисунок 4.7 Етапи перетворення зображення house

На зображенні присутні великі монотонні області, що залишає в частотному спектрі більше плавних ділянок. Це не дуже добре, тому що шум має таку статистичну характеристику, як випадковість, тому добре видаляється і обробляється різними фільтрами. Тому мітка із зображення з мережі «Facebook»

було прочитане з помилкою в 1 біт (3%). Якщо ми записували 16-бітове число, а решта 16 біт залишили для додавання надлишкової інформації для завадостійкого алгоритму кодування, то така помилка легко виправляється. Взагалі кажучи, якщо надлишкова інформація за обсягом дорівнює записується інформації, тобто надмірність $k/(i + k)$ дорівнює 50%, де k - кількість перевірочних біт, i - кількість інформаційних біт, то при використанні будь-якого оптимального алгоритму завадостійкого кодування (коди БЧХ, Ріда-Соломона), можна виправити до 50% помилок в переданому повідомленні. Таким чином, цей один втрачений біт легко відновлюється.

Третій тест - на зображенні lena (рисунок 4.8).

Рисунок 4.8 Етапи перетворення зображення lena

На кольоровому зображенні алгоритм показує аналогічні результати. Для зображення lena обидва варіанти, збережений з мережі та з месенджера, були прочитані без помилок. У розмірі, при цьому, зображення скоротилися у 8 разів.

Четвертий тест - на зображенні peppers (рисунок 4.9).

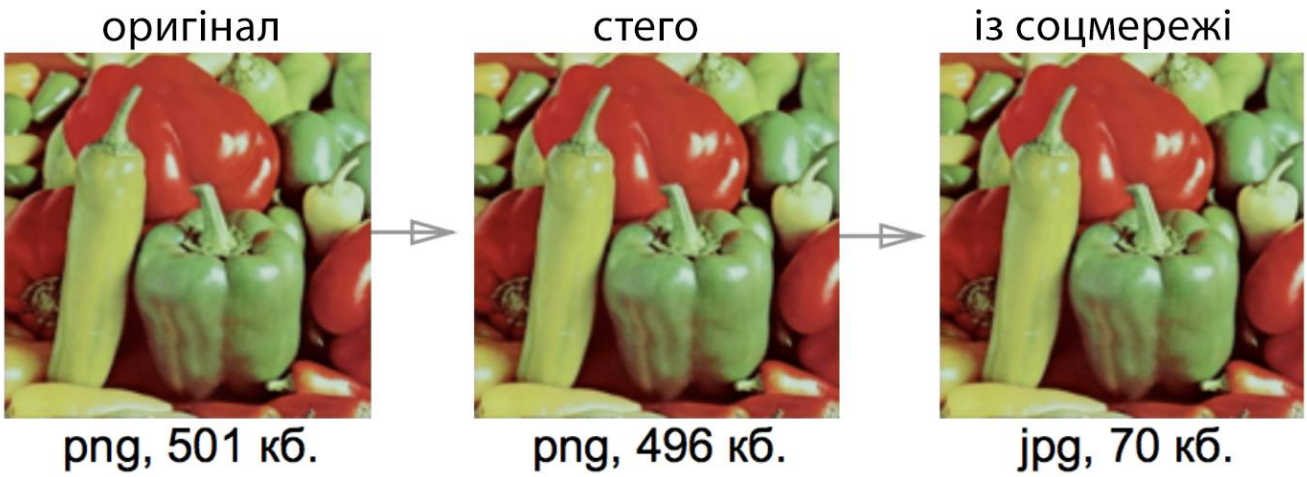

Рисунок 4.9 Етапи перетворення зображення pepper

Знову стикаємося з тим, що є великі монотонні області. Внаслідок цього, мітка з зображення з мережі «Facebook» була прочитана з 3 помилками (9%), вони можуть бути відновлені. Мітка з зображення з месенджера була прочитана без помилок.

Загально кажучи, виявилося, що месенджер зберігає фотографії та ілюстрації в більш високій якості, як наслідок мітки з усіх модельних зображень були прочитані без помилок [40].

4.3.2 Граничні значення

Можна зробити висновок про те, що алгоритм не зовсім придатний для використання на монотонних зображеннях - графіках, схемах, діаграмах. Відмінно працює на різнорідних зображеннях: монохромних, чорно білих та кольорових. Крім цього, слід враховувати наявність мінімально можливого розміру зображення для вбудовування мітки. Це залежить від параметрів вбудовування.

Наприклад, для параметрів:

- *Fs = 1920;*
- *bitrate = 16;*
- *N = 32;*
- *k = 5;*

мінімальний розмір контейнера складе 300x300 пікселів.

4.3.3 Результати

Алгоритм відповідає всім висунутим вимогам, а саме:

- виконується дуже швидко, в межах часу реакції користувача (до 1 с на зображеннях середніх розмірів);
- робастний до найбільш поширених атак;
- візуально оригінал і стего не відрізняються;
- вилучення мітки можливе без наявності контейнера-оригіналу.

4.4 Висновки до розділу 4

У четвертому розділі був описаний інструментарій для практичної реалізації алгоритму: обгрунтований вибір скриптової мови Python та основні бібліотеки, які необхідні для реалізації алгоритму.

Була наведена структура програми, що реалізує спроектований алгоритм та описані всі процедури мовою Python, що поетапно реалізують алгоритм. Цикл вбудовування описується наступними діями:

1) ряд y вихідного сигналу згортається в двомірний масив за допомогою розбиття ряду на відрізки заздалегідь визначеної довжини з урахуванням необхідної компактності мітки;

2) мітка вставляється в заздалегідь визначену позицію спектральної області шляхом заміни значень;

3) зворотне перетворення зображення з частотного спектру у видимий.

У четвертому розділі наводиться опис принципу тестування розробленого прототипу, метрики і безпосередньо результати.

Далі наводиться опис процесу читання мітки, який є складнішим процесом, ніж її генерація і вбудовування. Пов'язано це з тим, що сигнал потрібно демодулювати, буквально «вгадати» його значення в певний момент часу.

В наступному підрозділі був описаний процес тестування роботи алгоритму і його програмної реалізації: було вирішено використовувати середовище, близьке до

реальних випадків використання такого рішення, тому в якості «атакуючого» середовища була обрана соціальна мережа «Facebook» і популярний месенджер «Телеграм». Для тестування були обрані стандартні в області тестування алгоритмів для роботи з зображеннями ілюстрації: cameraman, house, lena, peppers.

Тести показали, що зображення з великими монотонними областями показують гірші результати і мітка читається з помилками. Також, виявилося, що месенджер зберігає фотографії та ілюстрації в більш високій якості, як наслідок мітки з усіх модельних зображень були прочитані без помилок.

Можна зробити висновок про те, що алгоритм не зовсім придатний для використання на монотонних зображеннях - графіках, схемах, діаграмах. Відмінно працює на різнорідних зображеннях: монохромних, чорно білих та кольорових. Крім цього, слід враховувати наявність мінімально можливого розміру зображення для вбудовування мітки.

ВИСНОВОК

У даній роботі був розроблений алгоритм швидкого робастного стегозапису для зображень, що дозволяє вилучати приховане повідомлення без наявності контейнера-оригіналу. Були розібрані існуючі методи цифрової стеганографії і проведено порівняльний аналіз, а також запропонований оригінальний спосіб швидкого стеганографічного перетворення.

В даному дослідженні було використано системний підхід щодо досягнення цілей роботи, міждисциплінарний підхід, а також методи логічного аналізу і синтезу.

Відповідно до поставленої мети були виконані наступні завдання:

- розглянуті основні механізми внесення цифрових водяних знаків у контент: внесення цифрових водяних знаків у зображення, звукові файли, відео;

- вироблені критерії для алгоритму вбудовування ЦВЗ;

- наведено порівняння методів вбудовування ЦВЗ;

- розроблений алгоритм захисту мультимедійних файлів на прикладі захисту зображень з подальшою можливістю розширення алгоритму для відеоконтенту;

- досліджено інструментарій для реалізації алгоритму та представлений прототип програмної реалізації алгоритму;

- реалізація алгоритму протестована в умовах, наближених до реальних.

Прототип був реалізований у вигляді скриптового Python-додатку, доступного для запуску з консольного рядка за допомогою інтерпретатора Python версії 2.7 і вище. Реалізоване ПО добре показало себе на різних тестах, в тому числі на наборах чорно-білих, монохромних і кольорових зображень.

Однак, були виявлені граничні випадки, використання алгоритму при яких є невиправданим. Наприклад, зображення з дуже великими монотонними областями. Серед таких: графіки, діаграми, скани тексту.

Проте, розроблений прототип і метод вбудовування готовий для впровадження в існуючі системи і може використовуватися як вбудований модуль,

на вхід якого подається зображення і повідомлення для запису, а на виході виходить зображення з прихованим повідомленням.

Розроблений алгоритм залишає великі можливості щодо його поліпшення та модифікації. наприклад:

- автоматичний пошук мітки в Фур'є-образі;
- тестування інших методів цифрової модуляції сигналу;
- автоматична компенсація ефекту повороту, розтягування/стиснення зображення;
- реалізація на компільовані мови типу C ++, що в рази прискорить виконання прогармми;
- застосування на відеоряді.

Автоматичний пошук мітки можливий подібно до того, як FM-радіоприймач налаштовується на «хорошу» хвилю. Крім цього, можуть бути знайдені інші методи модуляції, що показують кращі результати для граничних випадків.

Примітно те, що рішення задачі лежить на стику методів:

- цифрової стеганографії;
- цифрової обробки зображень;
- цифрової обробки сигналів;
- обробки експериментальних даних;
- завадостійкого кодування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Чарльз Маклеллан «Ключевые тенденции в сегменте SaaS в 2016 г.» [Електронний ресурс] Режим доступу: https://www.weekit.ru/its/article/detail.php? ID=187102.
2. Г.Ф. Конахович «Оценка эффективности методов стеганографического встраивания информации в спектральную область изображений» // АСУ и приборы автоматики. 2014. №168. С.59-63.
3. Грибунин В. Г., Оков И. Н., Туринцев И. В. Цифровая стеганография. – М.: Солон-Пресс, 2009. – 272 с.
4. Аграновский А. В., Балакин А. В., Грибунин В. Г. , Сапожников С. А. Стеганография, цифровые водяные знаки и стегоанализ. Монография. – М.: Вузовская книга, 2009. – 220 с.
5. Конахович Г.Ф. Комп’ютерна стеганографія. Теорія і практика / Г.Ф. Конахович, А. Ю. Пузиренко. – Київ: МК-Пресс, 2006. – 288 с.
6. Fridrich J. Applications of Data Hiding in Digital Images / J. Fridrich // Tutorial for the ISPAC S’98 Conferece. – Melbourne, Australia, 1999. – 33 p.
7. Kalker T. Considerations on watermarking security / T. Kalker //IEEE International Workshopon Multimedia Signal Processing. – Cannes (France), 2001. – P. 201–206.
8. Cayre F. Watermarking security: Theory and practice / F. Cayre, C. Fontaine, T. Furon //IEEE Trans. Signal Processing. – 2005. – Vol. 53. – P. 3976–3987.
9. Furon T. A survey of watermarking security / T. Furon // Proc. of Int. Work. on Digital Watermarking. – Siena (Italy). – 2005. – Vol. 3710. – P. 201–215.
10. Kuhn M.G., Stirmark”, available at – Security Group, ComputerLab, Cambridge University, UK (E -mail: mkuhn@acm.org), 1997. [Електронний ресурс] Режим доступу: http://www.cl.cam.ac.uk/~mgk25/stirmark/.
11. Craver S. Can Invisible Watermarks Resolve Rightful Ownerships? / S. Craver, N. Memon, B. Yeo, M. Yeung // International Society for Optics and Photonics. – Electronic Imaging'97, 1997. – P. 310 – 321.

12. Fridrich J. Robust digital watermarking based on key-dependent basis functions / J. Fridrich // The 2nd Information Hiding Workshop. – Portland, Oregon, April 15 – 17, 1998. – P. 168 – 190.
13. Хорошко В.О., Азаров О.Д., Шелест М.Є., Яремчук Ю.Є. Основи комп'ютерної стега- нографії: Навч. посіб. для студентів і аспірантів. — Вінниця: ВДТУ, 2003.
14. Генне О.В. Основные положения стеганографии.// Защита информации. №3, 2000.
15. N.F. Johnson, S. Jajodia, Steganalysis: The Investigation of Hidden Information, IEEE Information Technology Conference, Syracuse, New York, USA, Sept. 1st-3rd. 1998.
16. Барсуков В.С., Романцов А.П. Компьютерная стеганография: вчера, сегодня, завтра. Технологии информационной безопасности XXI века. [Електронний ресурс] Режим доступу: http://st.ess.ru/.
17. Пономарев К. И., Путилов Г. П. Стеганография: история и современные технологии. – М.: МИЭМ, 2009. – 70 с.
18. Алиев А. Т. Разработка моделей, методов и алгоритмов перспективных средств защиты информации в системах электронного документооборота на базе современных технологий скрытой связи: дис. … канд. техн. наук: 05.13.19. – Ростов-на-Дону: ЮФУ, 2008. – 216 с.
19. Мерзлякова Е. Ю. Построение стеганографических систем для растровых изображений, базирующихся на теоретико-информационных принципах.:дис. канд. техн. наук: 05.13.19. – Новосибирск: СибГУТИ, 2011. – 161 с.
20. Жгун Т. В. Модель скрытой передачи информации в каналах связи: дис. … канд. ф.-мат. наук: 05.13.18. – В. Новгород: НовГУ, 2003. – 187 с.
21. Абазина Е. С. Метод скрытой передачи информации с кодовым уплотнением в видеоданных // Информация и космос. – 2014. – № 4. – С. 33–38.

22. Цветков К. Ю, Федосеев В. Е., Коровин В. М., Абазина Е. С. Модель кодера скрытых каналов с кодовым уплотнением с использованием сигнальных последовательностей Франка-Уолша, Франка-Крестенсона // Труды НИИР. – 2015. – № 1. – С. 2–11.

23. Аграновский А. В., Балакин А. В., Грибунин В. Г. , Сапожников С. А. Стеганография, цифровые водяные знаки и стегоанализ. Монография. – М.: Вузовская книга, 2009. – 220 с.

24. Небаева К. А. Разработка необнаруживаемых стегосистем для каналов с шумом: дис. … канд. тех. наук: 05.12.13. – СПб.: СПбГУТ, 2014. – 176 с.

25. Коржик В. И., Небаева К. А. Основы стеганографии: учебно-методическое пособие по выполнению практических занятий.– СПб.: СПбГУТ, 2015. – 20 с.

26. Цветков К. Ю, Федосеев В. Е., Абазина Е. С. Применение двумерных нелинейных сигналов Франка-Уолша, Франка-Крестенсона в методе формирования скрытых каналов с кодовым уплотнением в структуре сжимаемых видеоданных // Наукоемкие технологии в космических исследованиях Земли. – 2013. – №. 4. – С. 32–40.

27. Юдін О.К., Зюбіна Р.В., Фролов О.В. Аналіз стеганографічних методів приховування інформаційних потоків у контейнери різних форматів. - Pedram // 31st International Conference on Distributed Computing Systems Workshops. Minneapolis, USA, 2011. P. 1-6.

28. Lecture Notes on Cryptography / Goldwasser S., Bellare M. – Cambridge, Massachusetts, 2001. – 283 p.

29. Абазина Е. С., Ерунов А. А. Результаты моделирования метода скрытой передачи информации с кодовым уплотнением в видеоданных // Системы управления, связи и безопасности. – 2015. – № 2. – С. 1–25. [Електронний ресурс] Режим доступу: http://journals.intelgr.com/sccs/archive/2015-02/01-Abazina.pdf.

30. Цветков К. Ю. Методы цифровой стеганографии и их приложения в сетях спутниковой связи // Сборник трудов II военно-научной конференции Космических войск. – СПб.: МО РФ, 2004. – Т. 2. – С. 344–349.
31. Цветков К. Ю., Ефимов С. Н., Осташов И. Т. Защита инфокоммуникационных систем и сетей специального назначения: учебное пособие. – СПб.: ВКА имени А.Ф. Можайского, 2010. – 160 с.
32. Коровин В. М. Метод и алгоритмы встраивания широкополосных цифровых водяных знаков в сжатые изображения // Сборник докладов международной научно-практической конференции «Особенности развития космической отрасли России и перспективы ее дальнейшей интеграции в систему международных экономических связей». – СПб.: БГТУ, 2007. – С. 175– 178.
33. Гонсалес Р., Вудс Р. Цифровая обработка изображений //М.: Техносфера. – 2012. – Т. 1104.
34. «Клієнт (інформатика)». [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Клієнт_(інформатика).
35. Братко Н., Тарасюк С. СЕРВЕР ОНОВЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ // ОКВНЗ «Інститут підприємництва «Стратегія», м. Жовті Води, Україна. [Електронний ресурс] – Режим доступу до ресурсу: http://confcontact.com/2013_04_04_zhv/25_Bratko.htm.
36. «Браузер». [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Браузер.
37. «Авторизація». [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Авторизація.
38. «Модуляція». [Електронний ресурс] Режим доступу: https://uk.wikipedia.org/wiki/Модуляція_(фізика).
39. Travis Fagerness «FSK Explained with Python» [Електронний ресурс] Режим доступу: https://www.allaboutcircuits.com/technical-articles/fsk-explained-with-python/.

40. Кулик М.В. Реалізація алгоритму робастного внесення цифрових водяних знаків мовою Python [Електронний Ресурс] / Кулик М.В. // Наука та освіта: ключові питання сучасності: зб. наук. праць «λόгοσ». - 2018. - Режим доступу:

41. Лісковський І.О., Кулик М.В. Реалізація стеганографічної системи для відео з використанням сингулярного розкладання. // Дванадцята міжнародна науково-технічна конференція «Проблеми телекомунікацій»; Десята міжнародна науково-технічна конференція студентів та аспірантів «Перспективи розвитку інформаційно-телекомунікаційних технологій та систем»: Матеріали конференції. К.: НТУУ "КПІ", 2018. – с. 288-290.

ДОДАТОК 1

Лістинг модуля кодування та вбудовування ЦВЗ

```
# -*- coding: utf-8 -*-
# Python 2.7
import sys
import struct
import inspect
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as signal
import scipy.signal.signaltools as sigtool
from PIL import Image # pip install pillow / PIL
from PIL import ImageFont
from PIL import ImageDraw
import warnings
warnings.filterwarnings('ignore')
# from reedsolo import *
np.set_printoptions(threshold=np.nan)

class SliceMaker(object):
     def __getitem__(self, item):
          return item
def fft(arr):
     fft = np.fft.fftshift(np.fft.fft2(arr)) /
float(np.size(arr))
     return fft
def ifft(arr):
     ifft = np.fft.ifft2(np.fft.ifftshift(arr)) *
float(np.size(arr))
     return ifft
def prepareImage(arr):
     return Image.fromarray(norm(arr))
def show(arr):
     prepareImage(arr).show()
```

```
def save(arr, name):
     prepareImage(arr).save(name)
def showSpectre(arr):
     arr = np.abs(arr)
     arr = 20 * (np.log(arr - np.amin(arr)) + 10)
     show(arr)
def norm(arr, shift=False):
     if shift:
          arr = arr - np.amin(arr)
return np.uint8(arr / float(np.amax(arr)) * 255.0)

#the following variables setup the system
Fc = 128 #simulate a carrier frequency of 128 hz
Fdev = 96 #frequency deviation, make higher than
bitrate 96 hz
bitrate = 16 #simulated bitrate of data (Fbit previously)
N = 32 #how many bits to send
A = 0.02 # amplitude 0.02
Fs = 1920 # 1920, sampling frequency for the
simulator, must be higher than twice the carrier frequency 1920
printbits = 16 #number of bits to print in plots
(N_prntbits)
signal_len = Fs/bitrate*N
stego_row = N*4 # N*4
stego_cols = signal_len/stego_row
numtaps = bitrate
show_plot = 0
x_offset = 10
y_offset = 10
repeats = 3 # 5
norm_threshold = 0
# binary = np.repeat([0,1,0,1,0,0,0,1,0,1,0,1,0,1,0,1], 2)
binary =
[
1,0,1,0,1,0,0,1,0,1,1,0,1,1,1,0,1,0,1,1,0,0,1,0,1,1,1,1,1,0,1,0
]
```

```
    data_in = np.asarray([int(x) for x in binary])
    t = np.arange(0,float(N)/float(bitrate),1/float(Fs),
    dtype=np.float)
    #extend the data_in to account for the bitrate and convert 0/1
    to frequency
    m = np.zeros(0).astype(float)
    for bit in data_in:
        if bit == 0:
            m=np.hstack((m,np.multiply(np.ones(Fs/bitrate),FcFdev))
        ) else:
            m=np.hstack((m,np.multiply(np.ones(Fs/
        bitrate),Fc+Fdev)))
    #calculate the output of the VCO
    y=np.zeros(0)
    y=A * np.cos(2*np.pi*np.multiply(m,t))
    y=y-np.amin(y)+0.00000001
    # make some noise
    sgn = np.zeros(stego_row).astype(float)
    for row in np.reshape(y, (stego_cols, stego_row)):
        sgn = np.vstack((sgn, np.tile(row, (repeats,1))))
    sgn = sgn[1:]
    sgnDims = np.shape(sgn)

    # source image
    imgSource = Image.open('test/house.png').convert("RGB")
    imgData = imgSource.getdata()
    imgArr    =    np.asarray(imgSource)    (hei,    wid,    chls)    =
np.shape(imgArr)
    overlay = sgn
    padTop, padLeft = tuple(np.subtract(np.shape(imgArr[..., 0]),
    np.shape(overlay)))
    overlay = np.pad(overlay,
        ((y_offset, padTop-y_offset),
        (padLeft-x_offset, x_offset)), 'constant',
        constant_values=0.0)
    overlay = overlay + np.fliplr(np.flipud(overlay))
```

```
slicing = SliceMaker()[y_offset:y_offset+sgnDims[0], -
(x_offset+sgnDims[1]):-x_offset]

channels = (
     np.where(overlay != 0, overlay, fft(imgArr[..., 0])),
     np.where(overlay != 0, overlay, fft(imgArr[..., 1])),
     np.where(overlay != 0, overlay, fft(imgArr[..., 2])),
)

channels = (
     np.clip(ifft(channels[0]), 0, 255),
     np.clip(ifft(channels[1]), 0, 255),
     np.clip(ifft(channels[2]), 0, 255),
)

Image.fromarray(np.uint8(np.dstack(channels))).save('results/
out.png')
```

ДОДАТОК 2

Лістинг модуля читання ЦВЗ

```
# -*- coding: utf-8 -*-
# Python 2.7
import sys
import struct
import inspect
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as signal
import scipy.signal.signaltools as sigtool
from PIL import Image # pip install pillow / PIL
from PIL import ImageFont
from PIL import ImageDraw
import warnings warnings.filterwarnings('ignore')
# from reedsolo import *
np.set_printoptions(threshold=np.nan)

class SliceMaker(object):
     def __getitem__(self, item):
          return item
def plot(arr, x=None):
     if x != None:
          if np.size(x) > 1:
               plt.plot(x, arr.T)
          else:
               plt.plot(range(x), arr.T)
     else:
          plt.plot(arr)
          plt.xlabel('x')
          plt.ylabel('y')
          plt.title('Plot')
          plt.grid(True)
          # plt.savefig("test.png")
```

```
          plt.show()
def shift(arr):
     return np.fft.fftshift(arr)
def fft(arr):
     fft = np.fft.fftshift(np.fft.fft2(arr)) /
float(np.size(arr))
     return fft
def ifft(arr):
     # ifft = np.fft.ifft2(arr)
     ifft = np.fft.ifft2(np.fft.ifftshift(arr)) *
float(np.size(arr))
     return ifft
def prepareImage(arr):
     return Image.fromarray(norm(arr))
def show(arr):
     prepareImage(arr).show()
def save(arr, name):
     prepareImage(arr).save(name)
def showSpectre(arr):
     arr = np.abs(arr)
     arr = 20 * (np.log(arr - np.amin(arr)) + 10)
     show(arr)
def merge(arr, mask, val):
     x, y = np.shape(arr)
     print x, y
     for i in range(x):
          for j in range(y):
               if mask[i, j] != 0:arr[i, j] = val if val > 0 else
               mask[i,j]
     return arr
def norm(arr, shift=False):
     if shift:
          arr = arr - np.amin(arr)
     return np.uint8(arr / float(np.amax(arr)) * 255.0)
def test():
     test = np.asarray(Image.open('results/
```

```
inverse.png').convert("L"))
     direct = fft(test)
     showSpectre(direct)
     exit()

#the following variables setup the system
bitrate = 16   #simulated bitrate of data (Fbit previosly)
N = 32         #how many bits to send
Fs = 1920           # 1920, sampling frequency for the simulator,
must be higher than twice the carrier frequency 1920
printbits = 16 #number of bits to print in plots
(N_prntbits)
signal_len = Fs/bitrate*N
stego_row = N*4 # N*4
stego_cols = signal_len/stego_row
numtaps = bitrate
x_offset = 10
y_offset = 10
repeats = 3 # 5
height = stego_cols * repeats
width = stego_row
slicing = SliceMaker()[y_offset:y_offset+height, -
(x_offset+width):-x_offset]

# binary = np.repeat([0,1,0,1,0,0,0,1,0,1,0,1,0,1,0,1], 2)
binary = [
1,0,1,0,1,0,0,1,0,1,1,0,1,1,1,0,1,0,1,1,0,0,1,0,1,1,1,1,1,0,1,0
]
data_in = np.asarray([int(x) for x in binary])

test = np.asarray(Image.open('results/input.jpg').convert("L"))
saved = fft(test)

restored = np.zeros(stego_row).astype(float)
received = saved[slicing]
```

```
for i in range(0, height/repeats):
     restored = np.vstack((restored,
np.average(received[i*repeats:i*repeats+repeats], axis=0)))
restored = restored[1:]

restored = np.abs(np.reshape(restored, (signal_len, )))
restored = restored - np.amin(restored)
y_diff = np.append(np.diff(restored,1), [.0])
#create an envelope detector and then low-pass filter
y_env = np.abs(sigtool.hilbert(y_diff))
h=signal.firwin(numtaps=numtaps, cutoff=bitrate*2, nyq=Fs/2)
# h=signal.firwin(numtaps, [63, 65], pass_zero=False, nyq=Fs/2)
y_filtered=signal.lfilter(h, 1.0, y_env)
# calculate the mean of the signal
mean = np.mean(y_filtered)
# if the mean of the bit period is higher than the mean, the
data is a 0
rx_data = []
# The decision is done at the center of the bit period
sampled_signal = y_filtered[Fs/bitrate/2:len(y_filtered):Fs/
bitrate]

for bit in sampled_signal:
     if bit < mean:
          rx_data.append(0)
     else:
          rx_data.append(1)

bit_error=0
for i in range(0,len(data_in)):
     if rx_data[i] != data_in[i]:
          bit_error+=1
error_percentage = (float(bit_error)/float(N)*100)

print 'errors = {1} ({0}%)'.format(error_percentage, bit_error)
```